

Improving Source-to-Destination Communication Schemes in Wide-Scale Cluster-Tree Wireless Sensor Networks

A thesis presented

by

Erico Meneses Leão

to

Doctoral Program in Informatics Engineering (ProDEI)

in partial fulfillment of the requirements for the degree

of Doctor in Informatics Engineering

Supervisor : Francisco Vasques - University of Porto

Co-supervisor : Paulo Portugal - University of Porto

September 2016

Porto, Portugal

This page was intentionally left in blank



UNIVERSIDADE
FEDERAL DO PIAUÍ



The work presented in this thesis was financially supported by the National Council for Scientific and Technological Development (CNPq-Brazil) with the grant GDE-201373/2012-2. The author would like to acknowledge the support from CNPq-Brazil and Federal University of Piauí, Teresina, Piauí, Brazil.

This page was intentionally left in blank

Abstract

Thesis Author
Erico Meneses Leão

Thesis Supervisor
Francisco Vasques

Thesis Co-Supervisor
Paulo Portugal

Improving Source-to-Destination Communication Schemes in Wide-Scale Cluster-Tree Wireless Sensor Networks

The use of wireless sensor network (WSN) based technologies is an attractive solution for wide-scale sensing applications such as environmental monitoring, precision agriculture and industrial automation. IEEE 802.15.4/ZigBee standards are the most used communication protocols for WSN technologies. Among the different network topologies proposed by these standards, the cluster-tree topology is pointed out as a suitable topology to support the implementation of wide-scale WSNs. The cluster-tree topology is a special peer-to-peer network, where nodes are grouped into clusters interconnected through their coordinator nodes. The design of this type of topologies encompasses relevant issues related to, for instance, network formation, beacon frame scheduling, network parameter configuration and multi-hop communication. Although presenting a series of advantages related to the duty-cycle operation and timing synchronisation, one of the most stringent limitations of cluster-tree networks is that all communication paths go through the PAN coordinator node (cluster-tree root). This communication behaviour results in a potentially higher number of hops along the path, higher energy consumption, higher end-to-end communication delays and higher network congestion around the PAN coordinator. These limitations encourage the design of new communication schemes.

Within this context, this thesis is focused on providing new efficient mechanisms to support source-to-destination message stream communication in IEEE 802.15.4/ZigBee cluster-tree WSNs for wide-scale deployments. Basically, two communication mechanisms are provided as main contributions: a set of *proportional Superframe Duration Allocation* (SDA) schemes and the *Alternative-Route Definition* (ARound) communication scheme. On one hand, SDA schemes provide guidelines to efficiently setup the communication structures of the coordinator nodes of cluster-tree networks, in order to improve the network throughput of the monitoring traffic (from sensor nodes towards the PAN coordinator). The main idea of these allocation schemes is to adequately allocate superframe durations, beacon intervals and buffer size values for cluster-head nodes, by considering well-defined protocol and timing network models and network features such as the message load imposed by child nodes and

the number of descendant nodes. The hypothesis is that the careful adjustment of communication structures may avoid typical problems of cluster-tree networks, such as: network congestion, higher end-to-end communication delays and discarded messages due to buffer overflows. Simulation assessments show that the use of proportional superframe duration allocation schemes clearly improves the network behaviour. On the other hand, the ARound communication scheme proposes the definition of alternative communication paths to support multi-hop message streams between source and destination nodes, by using cluster-tree nodes during their inactive periods. The main idea is to setup shortest inter-cluster paths through border nodes, in order to avoid the tree paths. The hypothesis is that, by setting-up alternative paths, the ARound communication scheme may improve performance metrics and save network resources. In fact, simulation assessments show that the proposed communication scheme can significantly decrease the end-to-end communication delay of message streams and the energy-consumption of network nodes. Also, the ARound scheme is able to reduce the network congestion, mainly near the PAN coordinator node.

Finally, this thesis also provides a new simulation model for wide-scale cluster-tree networks, that encompasses the most relevant communication mechanisms present in this type of networks, such as: random cluster-tree formation, hierarchical addressing, beacon scheduling schemes, superframe duration allocation schemes, direct and indirect data communication mechanisms, and different data traffic models. This simulation model was used for the assessment of the communication mechanisms proposed in this thesis, and will be shortly available for research community.

Keywords: *IEEE 802.15.4, ZigBee, Wireless Sensor Networks, Cluster-Tree, Wide-Scale, Message Streams, ARound, Allocation Scheme*

Resumo

Autor da Tese
Erico Meneses Leão

Orientador da Tese
Francisco Vasques

Co-Orientador da Tese
Paulo Portugal

Melhorias dos Esquemas de Comunicação Origem-para-Destino em Redes de Sensores sem Fio Cluster-Tree de Larga Escala

O uso de tecnologias baseadas em *Redes de Sensores Sem Fio* (RSSF) é uma solução atraente para aplicações de detecção em larga escala, tais como monitoramento ambiental, agricultura de precisão e automação industrial. As normas IEEE 802.15.4 e ZigBee são os mais utilizados protocolos de comunicação para tecnologias baseadas em RSSF. Dentre as diferentes topologias de rede propostas por essas normas, a topologia *cluster-tree* (árvore de agrupamentos) é apontada como uma topologia adequada para suportar a implementação de RSSFs em larga escala. A topologia *cluster-tree* é uma rede ponto-a-ponto especial, onde os nós de rede são agrupados em conjuntos interconectados através de seus nós coordenadores. O projeto deste tipo de topologia engloba questões relevantes relacionadas com, por exemplo, a formação de redes, escalonamento de quadros de *beacons* (balizas), configuração de parâmetros de rede e comunicação *multi-hop* (múltiplos saltos). Embora este tipo de topologia apresente uma série de vantagens relacionadas com a operação *duty-cycle* (ciclo de trabalho) e sincronização de tempo, uma das limitações mais rigorosas de redes *cluster-tree* é que todos os caminhos de comunicação fluem para o nó coordenador da PAN (nó raiz da rede). Este comportamento de comunicação resulta em um número potencialmente maior de saltos ao longo do caminho, maior consumo de energia, maiores atrasos de comunicação fim-a-fim e maiores congestionamentos de rede em torno do coordenador da PAN. Estas limitações encorajam a concepção de novos esquemas de comunicação.

Dentro deste contexto, esta tese está focada em fornecer novos mecanismos eficientes para suportar a comunicação de fluxo de mensagens da origem até o destino em Redes de Sensores Sem Fio *cluster-tree* baseadas nas normas IEEE 802.15.4 e ZigBee, para implantações em larga escala. Basicamente, dois mecanismos de comunicação são fornecidos como contribuições principais: o esquema de *Alocação Proporcional de Durações de Superframe* (SDA) e o esquema de *Definição de Rotas Alternativas* (ARound). Por um lado, o esquema SDA fornece diretrizes para configurar de forma eficiente as estruturas de comunicação para nós coordenadores de redes *cluster-tree*, a fim de melhorar

a taxa de transferência de rede do tráfego de monitoramento (a partir de nós sensores em direção ao coordenador da PAN). A ideia principal destes esquemas de alocação é alocar de forma adequada durações de superframe (super-quadros), intervalos de beacons e os tamanhos dos *buffers* para os nós coordenadores, considerando modelos de tempo e de protocolo bem definidos e características da rede, tais como a carga de mensagens imposta por nós filhos e o número de nós descendentes. A hipótese é que o ajuste cuidadoso das estruturas de comunicação pode evitar problemas típicos de redes *cluster-tree*, tais como: congestionamentos de rede, altos atrasos de comunicação fim-a-fim e o descarte de mensagens devido à *buffers* cheios. Avaliações através de simulação mostram que os esquemas de alocação de durações de *superframes* claramente melhoram o comportamento da rede. Por outro lado, o esquema de comunicação ARound propõe a definição de caminhos de comunicação alternativos para suportar fluxos de mensagens de múltiplos saltos entre um nó origem e um nó destino, usando os nós da rede *cluster-tree* durante os seus períodos de inatividade. A ideia principal é configurar caminhos mais curtos entre-clusters através de nós de borda, a fim de evitar os caminhos da árvore. A hipótese é que, através da criação de caminhos alternativos, o esquema de comunicação ARound possa melhorar métricas de desempenho e poupar recursos da rede. De facto, avaliações através de simulação mostram que o esquema de comunicação proposto pode diminuir significativamente os atrasos de comunicação fim-a-fim de fluxos de mensagens e o consumo de energia dos nós da rede. Além disso, o esquema ARound é capaz de reduzir o congestionamento de rede, principalmente nas proximidades do nó coordenador da PAN.

Finalmente, esta tese também fornece um novo modelo de simulação para redes *cluster-tree* em larga escala, que engloba os mais relevantes mecanismos de comunicação presentes neste tipo de rede, tais como: formação de redes *cluster-tree* de forma aleatória, endereçamento hierárquico, esquemas de escalonamento de beacons, esquemas de alocação de durações de *superframes*, mecanismos de comunicação de dados direta e indireta e diferentes modelos de tráfego de dados. Este modelo de simulação foi utilizado para a avaliação dos mecanismos de comunicação propostos nesta tese e irá ser disponibilizado brevemente para a comunidade científica.

Palavras-chave: IEEE 802.15.4, ZigBee, Rede de Sensores sem Fio, Cluster-Tree, Larga-Escala, Fluxo de Mensagens, ARound, Esquema de Alocação

*I would like to dedicate this doctoral
thesis to my beloved wife Marcela
and all my family.*

This page was intentionally left in blank

Acknowledgments

"I can do all this through him who gives me strength."

Philippians 4:13

Este curso de doutoramento já vem sendo idealizado desde 2008. Durante esses quatro anos, vivi momentos maravilhosos, mas também foram muitos desafios e muitas batalhas. Não foi fácil. A saudade da família, dos amigos e a distância de minha terra e de minha cultura foram obstáculos difíceis e muito desafiadores. Por isso, cada gesto, cada demonstração de afeto e carinho foram de extrema importância para minimizar a saudade e a distância. A conclusão dessa jornada é um momento muito especial na minha vida e, assim, só tenho a agradecer a todos que, de alguma forma, contribuíram para que esse sonho fosse realizado.

Primeiramente, gostaria de agradecer à Deus pelo dom da vida e por sempre me iluminar e guiar meus caminhos. Sem a Tua presença e Tuas bênçãos, nada disso seria possível.

À minha querida e amada esposa *Marcela Portela Leão*, meus mais sinceros agradecimentos. Não existem palavras nesse mundo que possam expressar o tão quanto sou grato e sou feliz por ter você em minha vida. Tudo isso só foi possível com a sua ajuda. Primeiramente, muito obrigado por compartilhar comigo todos os maravilhosos momentos que passamos durante todo esse processo. Entretanto, o meu mais profundo agradecimento por sempre estar ao meu lado durante os momentos mais difíceis e adversos durante todos esses anos (que não foram poucos). Você soube prover um ambiente seguro, confortável, acolhedor, que me acalmava e me tranquilizava nos momentos mais difíceis. Por isso, eu sou muito grato pelas palavras de carinho e de conforto, pelas sugestões e conselhos que você me deu nos momentos certos. Esta conquista é nossa! Sou muito orgulhoso da esposa que tenho. Como sempre digo: ao lado de um grande homem, sempre existe uma grande mulher! E você é e sempre será uma grande mulher! Te amo!

Aos meus pais, *Euclides Leão e Maria de Deus*, o meu amor eterno e incondicional. Vocês me ensinaram os principais valores da vida, a ser honesto, dedicado, lutador, perseverante. Desde cedo, vocês me estimularam a estudar e me mostraram que o caminho da vitória é glorioso, mas exige dedicação, trabalho e muito esforço. Hoje, eu sou quem eu sou, graças aos seus ensinamentos! Tenho muito orgulho de vocês, meus pais, e lhes dedico todas as minhas conquistas. Obrigado por serem os melhores pais do mundo, por quem tenho um amor infinito!

Aos meus queridos irmãos, *Edésio e Denise*, meu profundo agradecimento. Eu não poderia ter irmãos melhores. Como quaisquer irmãos, temos pensamentos contraditórios e opiniões divergentes! Entretanto, temos amor, união, carinho e afeto. Cresci numa família feliz e unida e sou muito grato por ela continuar sempre assim. E a família cresce cada vez mais. Por isso, também só tenho a agradecer aos meus cunhados *Glaxton e Keilla* e aos meus amados sobrinhos *Graziela, Davi e Victor*, que amo como filhos e que completam a minha querida família.

Aos meus demais queridos familiares, meu muito obrigado. Família é a base de tudo e agradeço todos os dias pela minha. Obrigado por torcerem por mim e por estarem sempre ao meu lado.

Aos meus sogros, *José Wellington e Conceição de Maria*, e meus cunhados *Maurício e Maysa*, meus sinceros agradecimentos. Obrigado por todas as palavras de carinho, sugestões e orações que tanto contribuíram e abençoaram essa conquista.

A melhor coisa na vida é ter bons amigos e eu sei que tenho os melhores e mais sinceros. Por uma questão de justiça e, até mesmo de espaço, prefiro não citar nomes. Só gostaria de agradecer, do fundo do meu coração, a todos vocês: muito obrigado por tudo. Pelas conversas via videoconferências, pelas mensagens, por levarem nossas fotos em circunstâncias especiais onde não pudemos estar presentes, pelas visitas aqui no Porto, pela torcida e pelas palavras de incentivo durante todo este período, por lembrarem da gente em tantos momentos, enfim, por todos os pequenos gestos que me fazem ter a certeza que tenho bons e grandes amigos. A saudade de todos foi grande e sou muito feliz em saber que irei reencontra-los em breve!

Ao *Francisco Vasques e Paulo Portugal*, meu orientador, co-orientador e amigos, sou eternamente grato pela forma magnífica como conduziram o meu trabalho. Agradeço por todas as produtivas reuniões, pelas brilhantes sugestões e ideias, pelas incansáveis correções dos trabalhos e por estarem sempre disponíveis para contribuírem ao máximo com a realização deste projeto. Além disso, pelas ótimas conversas e momentos descontraídos que tornaram mais leve esta longa caminhada. Meu muito obrigado e espero que possamos realizar muitos outros bons trabalhos ao longo de nossas jornadas. Gostaria de agradecer também aos colegas professores *Carlos Montez e Ricardo Moraes*, por suas relevantes contribuições, principalmente no tocante às sugestões nas publicações ao longo do doutoramento.

Aos colegas e amigos da Faculdade de Engenharia da Universidade do Porto (FEUP), meu agradecimento pelas conversas e discussões de trabalhos, especialmente pelas longas e divertidas conversas durante os vários cafés!

Não poderia deixar de agradecer à Universidade Federal do Piauí (UFPI), em especial a todos os professores do Departamento de Computação, por terem permitido o meu afastamento para a realização deste curso de doutoramento e por torcerem e acreditarem no meu trabalho. Espero poder contribuir ao máximo para o maior desenvolvimento da Computação na nossa Universidade e no nosso Estado.

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro para a realização deste curso de doutoramento no exterior (número de referência GDE-201373/2012-2).

À Portugal e, em especial, ao Porto, minha profunda gratidão. Aprendi a amar essa terra e fui muito feliz durante essa jornada. Aos poucos, mas bons amigos portugueses que fiz, o meu sincero agradecimento. Sentirei saudades e espero poder voltar em breve.

Meu muito obrigado a todos!

My heart is filled with gratitude!!

Erico Meneses Leão

Contents

Contents	xiii
List of Tables	xvii
List of Figures	xxi
Abbreviations and Acronyms	xxiii
1 Introduction	1
1.1 Research Context and Scope	1
1.1.1 WSN Applications	7
1.1.2 Research Problem	8
1.2 Research Objectives	10
1.3 Research Question and Hypothesis	11
1.4 Main Contribution and Outline	12
2 Supporting Real-Time Communication in Wide-Scale WSNs	15
2.1 Introduction	15
2.2 Background	16
2.3 Challenges of Real-Time Communication in IEEE 802.15.X	17
2.4 Wide-Scale and Real-Time Routing Protocols in WSN	19
2.5 Future Research Directions	22
2.6 Conclusion	22
3 A Simulation Model for Cluster-Tree WSNs	23
3.1 Abstract	23
3.2 Introduction	23
3.3 Overview of the IEEE 802.15.4/ZigBee set of standards	24
3.3.1 Cluster-tree network	25
3.4 Related work	26
3.5 CT-SIM WSN simulation model	29
3.5.1 Formation phase	30
3.5.2 Communication phase	34
3.5.3 Statistics and information collection	36
3.6 CT-SIM Simulation experiments	37

3.6.1	Node deployment	38
3.6.2	Cluster formation	39
3.6.3	Network addressing	39
3.6.4	Cluster scheduling	40
3.6.5	Superframe duration allocation	41
3.6.6	Communication phase	44
3.7	Conclusion and future considerations	46
4	Superframe Duration Allocation Schemes	49
4.1	Abstract	49
4.2	Introduction	50
4.2.1	Objectives and Contributions	51
4.2.2	Organisation of this Paper	52
4.3	Background	52
4.3.1	Cluster-tree Topologies	52
4.3.2	Related Works	53
4.4	System Model	56
4.5	Protocol Constraints	58
4.5.1	Length of the Beacon Interval	58
4.5.2	Dimensioning Buffer Sizes	61
4.6	Modelling the Transmission Time During the Superframe Duration	61
4.7	Timing Constraints	63
4.8	Superframe Duration Allocation Schemes	66
4.8.1	Proportional Allocation Scheme based on the Message Load	67
4.8.2	Proportional Allocation Scheme Based on the Number of Nodes	68
4.8.3	Illustrative Example	70
4.9	Simulation Assessment	73
4.9.1	Simulation Environment	73
4.9.2	Results and Discussion	78
4.10	Conclusion	86
4.10.1	Future Considerations	86
5	Alternative Path Communication Scheme	89
5.1	Abstract	89
5.2	Introduction	89

5.2.1	Objectives and Contributions	91
5.2.2	Organisation of this paper	91
5.3	Background	92
5.3.1	IEEE 802.15.4/ZigBee Cluster-tree Topology	92
5.3.2	Related Work	94
5.4	Alternative Paths for Message Streams in Cluster-Tree WSNs	98
5.4.1	Network Model	98
5.4.2	Alternative-Path Definition Algorithm	100
5.4.3	Alternative-Path Activation Algorithm	103
5.4.4	ARound Protocol Mechanisms	104
5.5	Simulation Assessment of the ARound Communication Scheme	110
5.5.1	Simulation Environment	110
5.5.2	Results and Discussion	114
5.6	Conclusions	120
5.6.1	Future Considerations	120
6	Conclusions and Future Work	121
6.1	Thesis Summary	121
6.2	Future Work	124
A	List of Publications	125
A.1	Submitted Manuscripts to Journals	125
A.2	Published Papers	125
A.2.1	Book Chapters	125
A.2.2	Publications in Conferences	125
	References	127

This page was intentionally left in blank

List of Tables

Table	Page
3.1 Comparison of the most used WSN simulators.	27
3.2 The main advantages of the proposed simulation model, compared with state-of-the-art simulation tools.	29
3.3 Parameters for the node deployment step.	31
3.4 Parameters for the cluster formation step.	32
3.5 Parameter used for the cluster scheduling step.	33
3.6 Parameters used for the superframe allocation scheme step.	35
3.7 Parameters used for the communication phase.	36
3.8 Informations about the simulations obtained through <i>CastaliaResults</i>	37
3.9 Information about the cluster-tree network formation.	39
3.10 List of addressing blocks of the cluster-heads, considering the random deployment.	40
3.11 Cluster scheduling for the random cluster-tree network experiment.	40
3.12 Static superframe duration allocation scheme for the random cluster-tree network experiment.	41
3.13 Proportional superframe duration allocation scheme for the random cluster-tree network experiment.	42
3.14 Size of internal buffers for the cluster-heads using different superframe duration allocation schemes.	42
3.15 Set of scenarios to evaluate the communication phase.	44
4.1 Superframe Durations for all Clusters.	71
4.2 Timing Constraint for all Message Streams.	73
4.3 Summarising the different Simulation Scenarios.	76
4.4 Simulation Configuration.	77
4.5 Information about the network formation.	78
4.6 Buffer size for cluster-heads (per depth).	78
4.7 Average Superframe Order values (per depth) for both the high load and low load zones (<i>conditioned Scenario</i>).	84
4.8 List of symbols, parameters and variables.	87
5.1 ACK frame types for the ARound communication scheme.	107
5.2 Simulation configuration.	113
5.3 Information about the network formation.	114

This page was intentionally left in blank

List of Figures

Figure	Page
1.1 An example of Wireless Sensor Network.	3
1.2 Star topology.	4
1.3 Peer-to-peer topology.	5
1.4 Cluster-tree topology.	6
1.5 IEEE 802.15.4 superframe structure	6
3.1 IEEE 802.15.4/ZigBee cluster-tree network.	25
3.2 IEEE 802.15.4 Superframe structure	26
3.3 CT-SIM implementation scheme.	30
3.4 Top-down cluster scheduling scheme.	33
3.5 Bottom-up cluster scheduling scheme.	33
3.6 Node deployment schemes.	38
3.7 Cluster-tree Network.	39
3.8 Average number of messages within the buffers of the cluster-heads (per depth), using top-down cluster scheduling.	43
3.9 Average number of messages within the buffers of the cluster-heads (per depth), using bottom-up cluster scheduling.	43
3.10 Number of discarded messages within the buffers of the cluster-heads (per depth), considering the top-down cluster scheduling.	44
3.11 Average end-to-end delay for the monitoring traffic under different number of sensing nodes.	45
3.12 Tree routing for the source-to-destination traffic.	46
3.13 Average Delay for the monitoring and source-to-destination traffics.	46
4.1 IEEE 802.15.4 Superframe structure.	53
4.2 IEEE 802.15.4/ZigBee cluster-tree network.	56
4.3 Top-down cluster active period scheduling.	57
4.4 Bottom-up cluster active period scheduling.	57
4.5 Length of beacon interval regarding message periodicities.	59

4.6	Beacon interval set to a value close to the shortest message period.	60
4.7	Beacon interval set to a value close to the sum of superframe durations.	61
4.8	Transmission time duration of a data frame.	62
4.9	Worst-case scenario for cluster-tree networks.	64
4.10	Local worst-case response time for a message stream.	64
4.11	Cluster-tree network composed of 6 cluster-heads and 12 leaf nodes.	70
4.12	Configuration of environment with one PAN coordinator and 200 randomly deployed nodes.	74
4.13	Configuration of the <i>unconditioned Scenario</i> : (a) Randomly distributed data rates; (b) Unconditioned cluster-tree network.	75
4.14	Configuration of the <i>conditioned Scenario</i> : (a) Distributed data rates in the two load zones; (b) Conditioned cluster-tree network.	76
4.15	Average discard message rate due to buffer overflows: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	79
4.16	Average message discard rate (per depth) due to buffer overflows for the STD-SDA scheme: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	80
4.17	Message loss rate for the SDA schemes: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	80
4.18	Beacon Interval and average sum of superframe durations for a network with longer beacon interval in the: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	81
4.19	Average end-to-end delay for a network with longer Beacon Interval in the: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	81
4.20	Average consumed energy for the SDA schemes: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	82
4.21	Average sum of superframe durations and beacon interval for the allocation schemes with adjustment of BI: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	82
4.22	Average end-to-end delay for the monitoring traffic with a shorter beacon interval: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	83
4.23	Average consumed energy for the SDA schemes with a shorter beacon interval: (a) <i>Unconditioned Scenario</i> ; (b) <i>Conditioned Scenario</i>	83
4.24	An example of Superframe Order (SO) configuration for all cluster-heads, using Load-SDA scheme (<i>conditioned Scenario</i>).	84
4.25	An example of Superframe Order (SO) configuration for all cluster-heads, using Nodes-SDA scheme (<i>conditioned Scenario</i>).	85
4.26	Number of schedulable simulations for the Load-SDA and Nodes-SDA schemes, considering different number of sensing nodes.	85
5.1	Cluster-tree networking topology.	92

5.2	IEEE 802.15.4 Superframe structure	93
5.3	Cluster-tree network obtained from simulations.	99
5.4	Cluster's active period scheduling along the time.	99
5.5	Duty cycle for the specific cluster 3, showing its active and inactive periods.	99
5.6	Standard cluster-tree path between the source and destination nodes.	100
5.7	Alternative ARound path between the source and destination nodes.	101
5.8	Applying the ARound-def Algorithm: (a) Selecting repeater nodes for the first pair of clusters; (b) Selecting repeater nodes for the second pair of clusters.	102
5.9	Format of the <i>ARound-req</i> frame.	106
5.10	Format of the <i>ARound-conf</i> frame.	106
5.11	Format of the <i>ARound acknowledgement command</i> frame.	107
5.12	Format of the <i>ARound auth</i> frame.	107
5.13	Message flow diagram for the <i>ARound negotiation phase</i>	108
5.14	Format of the <i>ARound closing command</i> frame.	109
5.15	Message flow diagram for the ARound closing mechanism.	109
5.16	Format of the <i>ARound hello command</i> frame.	109
5.17	Configuration of the <i>Scenario 1</i> : nodes 1 and 2 in adjacent quadrants.	111
5.18	Configuration of the <i>Scenario 2</i> : nodes 1 and 2 in opposite quadrants.	112
5.19	Average number of queued data packets in MAC buffers.	113
5.20	Information about source-to-destination message stream, considering the scenarios 1 and 2: (a) Average depth of nodes 1 and 2; (b) Number of hops for the cluster-tree and ARound paths.	115
5.21	Cluster-tree and ARound paths for one of physical topologies: (a) <i>Scenario 1</i> ; (b) <i>Scenario 2</i>	116
5.22	End-to-end Delay for ARound Approach vs. Standard Approach considering <i>Scenario 1</i>	116
5.23	End-to-end Delay for ARound Approach vs. Standard Approach considering <i>Scenario 2</i>	117
5.24	Packet Loss Rate for ARound Approach vs. Standard Approach for <i>Scenario 1</i>	118
5.25	Packet Loss Rate for ARound Approach vs. Standard Approach for <i>Scenario 2</i>	118
5.26	Impact of the <i>source-to-destination traffic</i> over the <i>background traffic</i> (<i>Scenario 1</i>).	119
5.27	Impact of the <i>source-to-destination data traffic</i> (ARound and the Cluster-tree approaches) over the <i>background data traffic</i> delay, considering the <i>Scenario 1</i>	119
5.28	Energy Consumption for the different node groups.	120

This page was intentionally left in blank

Abbreviations and Acronyms

ACK	Acknowledgement
ARound	Alternative-Route Definition
ARound-Act	Alternative-Path Activation Algorithm
ARound-Def	Alternative-Path Definition Algorithm
BAN	Body Area Network
BE	Backoff Exponent
BI	Beacon Interval
BO	Beacon Order
BS	Base Station
CAP	Contention Access Period
CCA	Clear Channel Assessment
CCH	Candidate Cluster-Head
CFP	Contention-Free Period
CH	Cluster-Head
CP	Communication Phase
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CT-SIM	Cluster-Tree Simulation Model
CW	Contention Window
DBR	Dynamic Bandwidth Re-allocation
DCR	Dynamic Cluster Re-ordering
EEHC	Efficient Heterogeneous Clustered Scheme
FFD	Full-Function Device
FP	Formation Phase
GAP	Global ARound Activity Period
GPS	Global Positioning System
GTS	Guaranteed Time Slots

GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
LAP	Local ARound Activity Period
LCM	Least Common Multiple
LEACH	Low-Energy Adaptive Clustering Hierarchy
LQI	Link Quality Indication
LR-WPAN	Low-Rate Wireless Personal Area Network
LTBS	Low-latency Two-way Beacon Scheduling
MAC	Medium Access Control
MCCT	Multi-Channel Cluster Tree
MEMS	Micro-Electro-Mechanical Systems
NED	Network Description Language
NICTA	National ICT Australia
NS	Network Simulator
PAN	Personal Area Network
PHY	PHYsical Layer
QoS	Quality of Service
RCPS/TC	Resource Constrained Project Scheduling with Temporal Constraints
RFD	Reduced-Function Device
RP	Repeater Nodes
RSSI	Received Signal Strength Indication
RTLD	Real-Time routing protocol with Load Distribution
SD	Superframe Duration
SDA	Superframe Duration Allocation
SDS	Superframe Duration Scheduling
SIC	Statistics and Information Collection
SO	Superframe Order
SUDAS	Superframe Duration Adjustment Scheme
TCBDGA	Tree-Cluster-Based Data-Gathering Algorithm

TDBS	Time Division Beacon frame Scheduling
TDCS	Time-Division Cluster Scheduling
TDMA	Time Division Multiple Access
WCRT	Worst-Case Response Time
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network

This page was intentionally left in blank

CHAPTER 1

Introduction

"The three great essentials to achieve anything worthwhile are, first, hard work; second, stick-to-itiveness; third, common sense."

Thomas A. Edison

THE RESEARCH WORK presented in this Thesis intends to be a contribution to the advance of the state-of-the-art of wide-scale cluster-tree Wireless Sensor Networks. In this chapter, firstly the research context, scope and motivation for this work are presented. Following, the research problem is stated and the key contributions of this research work are outlined. Finally, the document structure is presented.

1.1 Research Context and Scope

Nowadays, automation activities, combined with the continuous technological advance, are essential for the competitiveness increase in all industrial sectors. Indeed, in the current globalised world, it is increasingly demanded the development of new technologies, in order to increase productivity and efficiency in all industrial sectors [1]. With the advent of new wireless technologies, an increasing number of applications have focused in wireless connectivity, including industrial automation, distributed control systems and general networked embedded systems [2]. In fact, wireless communication systems provide a series of advantages, when compared with traditional wired communication technologies, such as: reduced deployment and management costs, mobility and the possibility of deployment in harsh environments, where the human intervention is limited [3, 4].

Within this context, the use of *Wireless Sensor Network* (WSN) based technologies is an attractive research field, due to the wide range of application domains, such as: industrial and home automation, military applications, environment and building monitoring, target tracking, automotive systems, health monitoring, smart cities, and agricultural monitoring [5–8]. Note that, currently, WSNs are being extended and deployed to practically all human activities. Wireless sensor Networks (WSNs) are special *ad hoc* networks composed of a varying number of specific devices, commonly called *sensor nodes*. Sensor nodes are low-cost, low-power, and low-rate wireless devices with capabilities for sensing,

processing, and communicating data upon a given monitored environment. Basically, sensor nodes are composed of the following components:

- Sensor circuitry: used to gather information/data about physical phenomena of monitored environments, such as: temperature, pressure, humidity, motion, light, and others;
- Processing unit: used to process sensed data and to perform additional computing activities;
- Internal memory: used to locally store sensed and program data;
- Communication module: used to communicate with neighbouring nodes, in order to transmit and receive data;
- Energy supply unit: power source used to provide energy to sensor nodes (batteries and/or energy harvesting mechanisms are commonly used).

Figure 1.1 illustrates a typical WSN scenario. Sensor nodes are deployed along of a specific environment, in order to monitor a set of given physical variables. Depending on the application requirements, the deployment of nodes can be structured (predefined) or unstructured (randomised) [9, 10]. In a structured deployment, sensor nodes are strategically deployed along the monitored environment, which enables a planned network construction and save resources [10]. This type of deployment is suitable for known environments, reducing deployment and management costs and enabling a total environment coverage with a small number of sensor nodes. In turn, there are many environments where the predefined deployment is hard or even impossible. In this case, an unstructured deployment becomes necessary. In the unstructured deployment, sensor nodes are deployed using a randomised strategy. This type of deployment is suitable for harsh environments, where the human access is limited or unreachable and wired systems are unsuitable [8, 9]. This deployment is also commonly used in wide-scale¹ environments.

After being deployed, sensor nodes are responsible for sensing physical phenomena around their coverage area and reporting these data to special nodes, called *sink nodes* (also known as *base station* - BS). As each sensor obtains limited local information, sensor nodes should self-organise into an ad hoc network and collaboratively work towards a common task [2, 6, 8]. The sink node is responsible for receiving the information gathered by sensor nodes and acts as interface with end users or gateway for other networks. According to Kumar et al. [7], WSNs can be designed with one or more sink nodes, depending on the application requirements.

It can be also observed in Figure 1.1, that some nodes may not directly communicate with sink nodes. In this case, sensor nodes can be used as *intermediate nodes* (or *repeater nodes*) for more distant nodes. This assumption often occurs in *wide-scale* environments, where intermediate nodes are responsible for relaying data from distant nodes. This type of networks is frequently called of multihop networks [2, 4, 6, 9, 11].

¹In this Thesis, we adopt the *wide-scale* term to refer geographically wide deployment environments, as the *large-scale* term commonly used in WSN Literature is often related to highly-dense WSNs, with a large number of sensor nodes in a small scale environment.

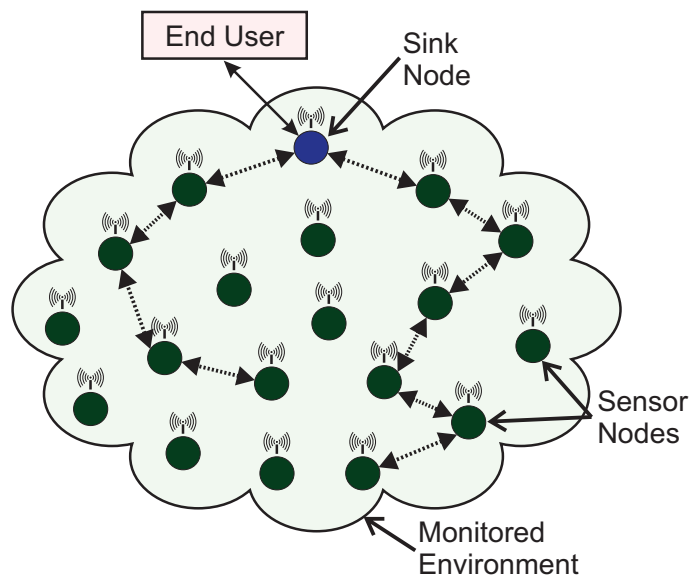


Figure 1.1: An example of Wireless Sensor Network.

Following the same reasoning, wireless devices (nodes) may also be equipped with special transducers called *actuators*, with capability to actuate upon the monitored environment. In this case, these nodes are called *sensor/actuator nodes* and these networks are often referred as *Wireless Sensor and Actuator Networks (WSANs)* [5, 12]. According to Akyildiz and Kasimoglu [12], WSANs are composed of a large number of sensor nodes that are responsible for gathering information about the environment, while a smaller number of actuator nodes are responsible to perform appropriate actions based on these observations. With this, WSN application domains are even greater.

WSNs present a set of advantages over traditional *ad hoc* networks, within which we can highlight: flexibility, mobility, autonomous and collaborative operation. However, unlike traditional *ad hoc* networks, WSNs impose a set of constraints related to the resource limitations of sensor nodes (energy, processing and storing) and inherent features of low-power wireless communication. Moreover, the monitored environment also imposes a set of constraints (size, obstacles, hazardous area, and others), which must be considered in the design of WSNs. Within this context, the design of WSNs is a challenging task, due to the large number of constraints that need to be simultaneously satisfied. This way, based on WSN literature, we can highlight a set of important issues that must be taken into account when designing new WSN approaches, protocols or algorithms, such as:

- **Energy-Efficiency:** sensor nodes are commonly powered by limited batteries (energy supply). In this way, the energy consumption is one of most important requirements when designing WSNs. Moreover, in hazardous environments, the replacement of batteries can be impossible or impractical. Thus, energy-efficiency operation is one of the most important issue in WSNs [6, 13].
- **Self-organisation:** hazardous or wide-scale environments may require randomised deployment. In this case, sensor nodes must have self-organisation capabilities without the human intervention, in order to build adequate communication network topologies [14, 15].

- Scalability: depending on the application, WSNs may be composed of a large number of sensor nodes (tens, hundreds or thousands) along a wide-scale environment, which imposes a strict constraint, regarding to the design of the network topology and communication protocols. Thus, scalability becomes an important requirement to design WSN applications [4, 16]
- Real-time: critical applications impose time-sensitive requirements, where data transmissions have real-time constraints. In this case, the design of WSNs must ensure individual message deadlines [17].

In recent years, several specifications were released by standardisation bodies and industrial alliances, that are suitable for building WSNs with different characteristics and requirements for the most diverse application domains. Within this context, it can be highlighted the following specifications: IEEE 802.15.4 [18], IEEE 802.15.4e [19], IEEE 802.15.5 [20], ZigBee [21], WirelessHart [22], and 6LoWPAN [23], being the IEEE 802.15.4/ZigBee set of standards the most used wireless technologies to build Wireless Sensor Networks.

The IEEE 802.15.4-2015 [18] standard defines a PHY layer (PHY) and Medium Access Control (MAC) sublayers for Low-Rate Wireless Personal Area Network (LR-WPAN), focusing on short-range operation, low-data rate, energy-efficiency and low-cost implementations. In turn, ZigBee [21] specifies the upper layers (Network and Application) over the IEEE 802.15.4 protocol stack.

According to the application requirements, the IEEE 802.15.4/ZigBee standards basically define two types of network topologies: star and peer-to-peer. In the star topology, all sensor devices directly communicate with a unique node, named *PAN coordinator* (centralised paradigm). The PAN coordinator is the primary controller and is responsible to initiate, terminate, and route communication around the network. Figure 1.2 illustrates an example of a star topology.

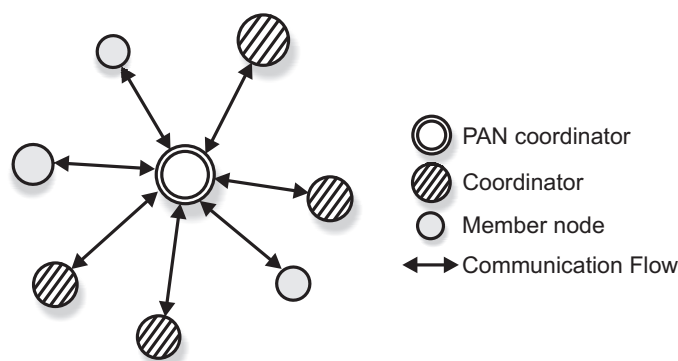


Figure 1.2: Star topology.

Despite its simplicity, this topology mainly fails regarding to scalability, because its coverage is bounded by the transmission range of its member nodes, which prevents the building of wide-scale WSNs.

In the peer-to-peer topology, each device can communicate with any other node, as long as they are in the communication range of one another. This way, this topology allows to

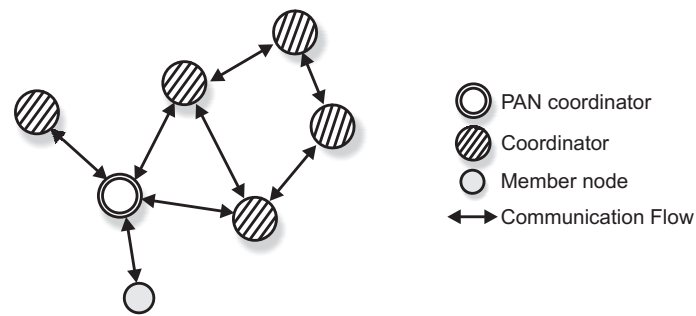


Figure 1.3: Peer-to-peer topology.

implement more complex network formations, such as mesh and cluster-tree topologies. Figure 1.3 illustrates an example of peer-to-peer topology.

The mesh networking topology uses a decentralised communication paradigm and allows multihop routing through neighbouring nodes, where sensor nodes can act as routers, forwarding packets from other nodes that are not within direct transmission ranges of their destination nodes [24]. Mesh topologies provide network flexibility and lower complexity, besides routing redundancy and good scalability [25, 26]. However, this topology does not provide explicit timing synchronisation mechanisms, which would allow nodes to enter into low power mode [27, 28], to increase the network lifetime. Moreover, an additional complexity is added in order to ensure end-to-end connectivity [26]. These features are not desirable for typical WSN-based monitoring applications, where energy-efficiency and lower complexity are crucial issues.

The cluster-tree networking topology is a special case of peer-to-peer topologies. Cluster-tree is pointed out as one of the most suitable topologies for building wide-scale WSNs [29, 30]. In this type of topology, nodes are grouped into *clusters*, being coordinated by a node called Cluster-Head (CH). CHs are responsible to provide synchronisation for its associated nodes and centralize all intra-cluster communication. The cluster-tree network formation is started by a unique node called PAN coordinator, which acts as coordinator for the network and is responsible for all management activities. According to the IEEE 802.15.4 standard, a single cluster can be considered the simplest case of a cluster-tree network. Several neighbouring clusters can be interconnected through their coordinator nodes (cluster-heads), forming a multicluster hierarchical network structure, in order to increase the network size and to enable wide-scale deployments. Figure 1.4 illustrates an example of an IEEE 802.15.4/ZigBee cluster-tree network.

When using cluster-tree topologies, the network operates in beacon-enabled mode. In this mode, all communication activities are performed according to a structure called superframe. A superframe is bounded by beacon frames that are periodically transmitted by coordinator nodes (cluster-heads). Beacons are used to synchronize the associated nodes and to describe the superframe structure. A superframe structure is composed of two parts: active and inactive periods. In the inactive period, nodes can enter in low power mode to save energy. In the active period, communication between cluster nodes can be performed. The active part comprises two periods: contention access period (CAP) and contention-free period (CFP).

In the CAP period, communicating nodes must contend for the access to the wireless channel using a slotted CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance)

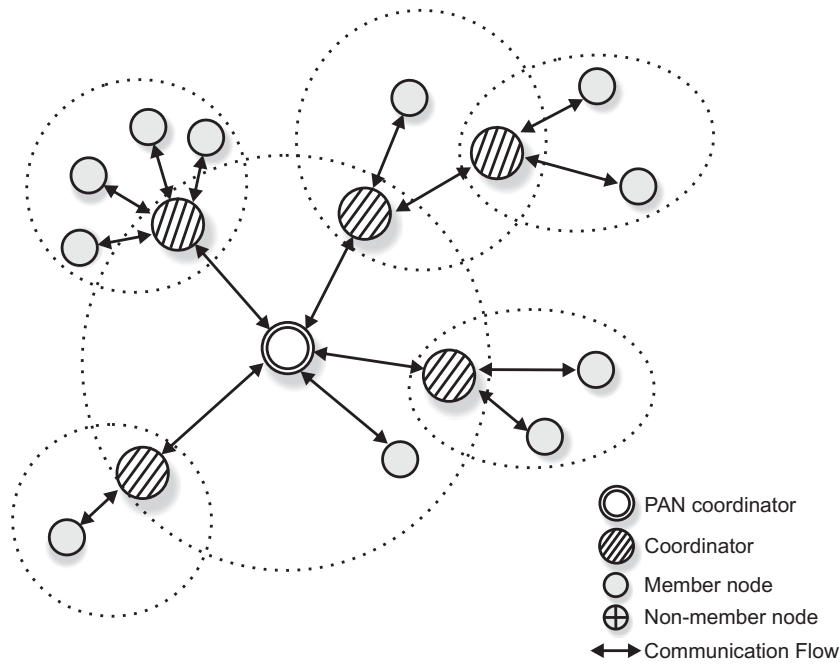


Figure 1.4: Cluster-tree topology.

channel access mechanism. For applications that require low latency or specific bandwidth, the CFP period is introduced. In the CFP period, coordinator nodes can allocate *guaranteed time slots* (GTS) for specific devices. In these slots, nodes can transmit data frames without contending for the wireless channel. Coordinator nodes can allocate up to seven GTSs to their sensor nodes, and each sensor node can use more than one GTS.

The superframe structure is defined by *macBeaconOrder* (BO) and *macSuperframeOrder* (SO) parameters. These parameters define the Beacon Interval (BI) and the Superframe Duration (SD), respectively. BI defines the interval at which the coordinator must transmit its beacon frames. SD defines the length of the active portion of the superframe. Figure 1.5 illustrates the superframe structure and describes the SD and BI values.

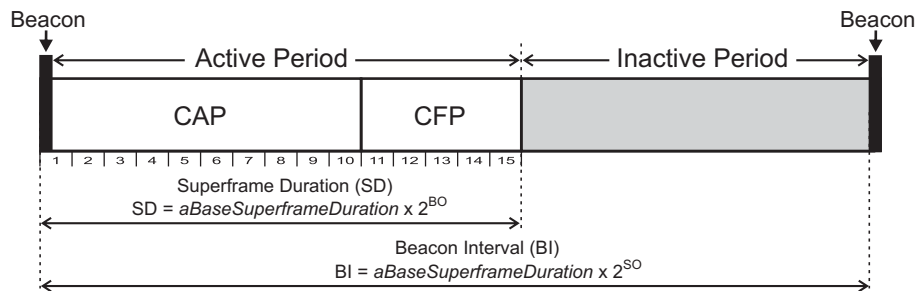


Figure 1.5: IEEE 802.15.4 superframe structure

Although IEEE 802.15.4 considers cluster-tree topologies, this standard does not provide explicit mechanisms to deal with cluster-tree networks. For this purpose, the ZigBee specification defines the network and application layers over the IEEE 802.15.4 protocol stack, providing mechanisms that enable the construction of cluster-tree networks, such as: network formation and association rules, hierarchical addressing and tree-based routing. Nevertheless, IEEE 802.15.4/ZigBee-based cluster-tree WSNs still impose several

challenging issues, encouraging researchers to develop new protocols and algorithms to solve open research questions of typical wide-scale deployments of WSNs.

1.1.1 WSN Applications

In 2002, Akyildiz et al. [9] envisaged that WSNs would be integral part of people's live, even more than personal computers. In fact, this assertion has become a reality. Currently, WSNs have been adopted in a growing number of application domains, from the simplest one until the most complex and hostile application scenarios.

This way, a few examples of wide-scale WSN applications may be summarised as follows:

1. **Industrial Applications.** Currently, WSNs are widely used in the industrial domain, such as: factory and process automation, coordination among robots, target tracking, and industrial environment monitoring [2, 31]. According to Kumar et al. [7], by using WSNs to monitor and control industrial environments, the human presence could be eliminated in dangerous and hostile places. Also, the deployment and management costs can be reduced by implementing wireless technologies. Despite the advantages provided by the use of WSNs, WSN-based industrial applications impose a set of unique features and issues that must be considered when designing new protocols and approaches, such as: tight real-time requirements, reliability, fault tolerance, security and privacy, scalability, energy efficiency operation, and hostile and noisy environments [1, 2]. These issues require industrial, sensor and networking expertise.
2. **Military Applications.** WSNs can be used in military applications for different purposes, in order to collect relevant information about battlefields, and to allow strategic decision-making [32]. According to Akyildiz et al. [9], WSNs can be an integral part of military *command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting* systems. A few examples of the WSN aplicability for military purposes are: enemy force detection, friendly force monitoring, soldier health monitoring, equipment monitoring, battlefield surveillance, targeting, attack detection, secure communication, among others [9, 32, 33]. Lee et al. [34] pointed out some of the most relevant requirements imposed by WSN-based military applications: security, fault tolerance, scalability, connectivity, energy efficiency operation and reliability.
3. **Environmental Monitoring.** Environmental monitoring is one the most typical WSN applications. Basically, a large number of sensor nodes may be deployed in a wide-scale area, in order to periodically monitor and sense detailed information about various physical variables, such as temperature, humidity, light, sound and pressure, and to send these information for base stations. WSNs can be used for a wide variety of environmental applications, such as: animals tracking and monitoring, environment monitoring, forest monitoring, irrigation, chemical/biological detection, precision agriculture, geophysical monitoring, fire detection, meteorological monitoring, among other [9, 35, 36]. Several environmental monitoring applications can be designed, imposing different challenges that are dependent of the monitored environment. For example, the requirements imposed by a volcanic monitoring (harsh environmental

conditions) are often most restrictive than an agricultural monitoring. In general, in order to develop environmental monitoring applications, the following requirements are considered: autonomous operation, energy efficiency operation, self-organisation, reliability, robustness and flexibility [36].

4. **Precision Agriculture.** Precision agriculture can be considered a special case of environmental monitoring, related to efficiently monitor farming environments. In this context, WSN-based technologies are responsible for automating information collection processes about the crop production. Based on these collected information, control and actuating strategies (for instance, irrigation, temperature control, fertilisation, and pest control) can be performed, in order to maximise the crop yield and quality, and to optimise the environment resources [37–39]. Therefore, the development of efficient data communication mechanisms are essential for obtain appropriate results and to maximise the agricultural production.
5. **Healthcare Monitoring.** WSN has become an attractive technology to monitor the health care of patients and overall people, which may be used for the following purposes: complete patient health monitoring, diagnostic, drug administration, doctor and patient tracking, among them. [9, 10, 32]. According to Ammari et al. [32], sensor nodes can be used to periodically collect information about the vital signals of patients and to send them to medical professional in order to monitor or actuate in emergency case.
6. **Urban Applications.** This application class is becoming very attractive, because it encompasses a series of useful applications, considering the metropolitan environments, such as: structural and environmental monitoring, power system applications, transportation and roadside applications, among others [40]. For example, Nellore and Hancke [41] highlight WSN-based urban traffic management systems, that are responsible to collect real-time data about the vehicular traffic, send them to specific processing units and to process these information in order to implement adequate actions. These systems can provide an efficient real-time vehicular traffic management, acting as a tool to avoid congestion, prioritize emergency traffics and for safety purposes.

In general, the communication schemes proposed in this Thesis can be easily applied in most application domains cited above. Nevertheless, we envisage the use of the proposed schemes in real world applications such as precision agriculture and environmental monitoring applications, which require the design of efficient communication mechanisms to deal with the high data traffic generated by a large number of widely deployed sensor nodes.

1.1.2 Research Problem

As previously discussed in Section 1.1, the use of cluster-tree topologies has become an attractive choice to design efficient approaches for WSN-based wide-scale applications. On one hand, the recent advances in Micro-Electro-Mechanical Systems (MEMS) technologies [9, 10], microprocessing power and low-power radio technologies [6] have enabled the development of tiny and low-cost wireless sensor devices increasingly robust, with more storage, processing, and energy capabilities, allowing the design of wide-scale

applications composed of massive number of sensor devices [8]. On the other hand, clustering-based hierarchical topologies (as the cluster-tree topology) present a series of benefits to build wide-scale WSNs, such as: advantages of beacon mode (that provides timing synchronisation, collision-free time slots, and duty-cycle operation), distributed network control, energy-efficient operation, reduced network contention, and the possibility of implementing data aggregation or fusion mechanisms [29].

Nevertheless, IEEE 802.15.4/ZigBee-based cluster-tree networks still present some relevant limitations and challenges, especially with regard the design of efficient approaches for wide-scale applications with real-time and energy efficiency requirements [42, 43]. Therefore, within the context of this Thesis, we have identified the following four relevant research problems, that are enumerated and detailed as follows:

1. **Cluster-tree Network formation.** The cluster-tree network formation process involves a set of procedures performed by sensor nodes, in order to self-organise into clusters, such as: the selection of a communication channel, association procedure, admission control and the election of coordinator nodes. According to Cuomo et al. [44], the topology formation is an important phase, because it can influence several network performance parameters, such as: energy consumption, network lifetime, delays and the coverage of environment. The resulting network topology can also impact the information collection activities performed by a cluster-tree WSN. Moreover, other important research problem regarding to cluster-tree formation is related to *orphan nodes* [45]. Due to common inefficiencies of available formation protocols, nodes may be prevented to join the network, leading the existence of orphan nodes. According to Kim et al. [46], a large number of coordinator nodes may improve the network connectivity and coverage; however, it will also increase the energy consumption and message delays. On the other hand, a smaller number of coordinator nodes may generate a large number of orphan nodes. Considering this topic, relevant contributions were proposed by Cuomo et al. [47] and Bandara et al. [48].
2. **Beacon scheduling.** One the most studied problems in cluster-tree literature is the beacon frame scheduling [26, 49]. In cluster-tree networks, cluster-heads periodically send beacon frames to their associated nodes. However, sending beacon frames without any synchronisation mechanism may result in serious collision problems [50]. Two types of beacon collisions are related: direct and indirect beacon collisions [26]. In a direct beacon collision, two or more cluster-heads in the same transmission range transmit their beacon frames at the same time, leading to a collision. In the indirect beacon collision, two or more coordinators, which cannot directly hear each other, transmit their beacon frames at the same time. However, as they have overlapped transmission ranges, this transmission may generate collisions (hidden-node problem). Moreover, beacon frames can also collide with data frames. The beacon collisions may generate serious problems for cluster-tree networks, as desynchronisation of nodes and partition of the network. Therefore, synchronization mechanisms must be provided in order to avoid the beacon frame collision. Ahmad and Hanzalek [51] define the beacon scheduling problem as: *"the construction of collision free and periodic cluster schedule specifying which cluster will be active at which time and which slot will be assigned to which data flow"*. Regarding the beacon frame scheduling, some relevant works have been presented in [26, 49, 52–54].

3. **Network parameter configuration.** Another important research problem is related to the MAC parameters configuration. There are several MAC parameters that have a direct impact over the performance of IEEE 802.15.4 protocol. On one hand, we have the parameters that are related to the CSMA-CA medium access protocol [28, 55]; on the other hand, MAC parameters that define the superframe structure [56]. Within the context of the superframe duration configuration, the definition of adequate MAC protocol parameters may provide a better network performance and avoid problems, such as: packet drops due to buffer overflows, congestion and high end-to-end delivery delays, which could lead to the undesired operation of the network. Thus, it is necessary to provide adequate guidelines for setting-up the MAC configuration parameters in order to build efficient wide-scale cluster-tree WSNs. Examples of research works on communication structure configuration can be found in [42, 56, 57].
4. **Efficient and alternative communication schemes.** In cluster-tree networks, all communication are performed by tree-based routing, i.e., data messages will follow paths defined by parent-child relationships. This way, cluster-tree paths assume that all communication go toward the PAN coordinator (sink) node along the tree path, which may impose serious problems, such as: congestion, delays, buffer overflows and higher energy waste (mainly coordinator nodes). Moreover, this type of static routing does not provide any alternative mechanism to forward messages among neighbouring clusters without using tree paths. In order to overcome these weaknesses, the careful design of new opportunistic communication approaches is desirable, thus guaranteeing the most part of the requirements imposed by different applications. Within this context, we highlight the research works proposed in [16, 43, 58, 59].

The research work presented in this Thesis is mainly focused in the following research issues: *Efficient and Alternative Communication Schemes* and *Network Parameter Configuration*.

1.2 Research Objectives

Due to some of the inherent features of wide-scale WSNs, mainly related to the large number of deployed low-power and low-rate wireless devices and to their memory-constrained characteristics, most of the proposed communication protocols present serious weaknesses [30, 60, 61]. According to Chakchouk [62], traditional routing protocols are not suitable or even unfeasible for wide-scale WSNs. The constraints imposed by WSNs and the limitations of traditional communication schemes have motivated the design of new communication paradigms and approaches, that may efficiently satisfy the requirements imposed by wide-scale WSN applications [63]. This assumption has been identified as the main motivation of this thesis work.

Within this context, **the main objective of this Thesis is to research and develop a set of novel mechanisms to improve source-to-destination message stream communication in IEEE 802.15.4/ZigBee Cluster-Tree WSNs**, specially applied to wide-scale deployments. This Thesis encompasses two types of source-to-destination communication: data communications from sensor nodes toward sink node (commonly the

PAN coordinator), and specific source-to-destination data communication between source and destination nodes.

Therefore, the goals of this thesis work has been addressed as follows:

1. The formulation, implementation and assessment of a novel communication scheme to support peer-to-peer communication in cluster-tree WSNs, avoiding to relay messages through the PAN coordinator (root node), called *Alternative-Route Definition* (ARoundD) scheme. The proposed scheme defines alternative end-to-end communication paths to deal with message streams in a collision-free IEEE 802.15.4 cluster-tree WSNs, without interfering with the pre-defined cluster-tree communication, by using relay/router nodes during their inactive periods, in order to improve several performance metrics of cluster-tree networks.
2. The formulation, implementation and assessment of a holistic approach to adequately allocate superframe durations to multiple clusters in a wide-scale WSN, called *proportional Superframe Duration Allocation* (SDA) scheme. Based on defined protocol and timing models for cluster-tree WSNs, the proposed SDA schemes define a set of specific guidelines for the network configuration, in order to improve the network throughput in wide-scale WSNs and to avoid typical problems of these networks, such as: congestion, high end-to-end communication delays and dropped messages due to buffer overflows.

1.3 Research Question and Hypothesis

This Thesis has focused on the following fundamental research questions and hypotheses:

How to define alternative communication paths to support on-demand message streams between source and destination nodes, using available network resources of sensor nodes during their inactive periods in IEEE 802.15.4/ZigBee cluster-tree WSNs? For this question, the research hypothesis is that, considering a previously scheduled cluster-tree WSN, the definition of a set of well-defined structures and efficient algorithms allows to build an alternative communication structure that may be used to support message-stream communication, guaranteeing a minimum level of quality of service, which can improve performance metrics such as: end-to-end communication delays, successfully data delivery, overall energy consumption, and network lifetime.

How to improve the traffic communication generated by sensor nodes in IEEE 802.15.4/ZigBee cluster-tree WSNs, avoiding known problems, such as congestion, high end-to-end delays and buffer overflows? For this question, the research hypothesis is that, considering adequate allocation schemes that proportionally set the superframe durations, beacon intervals and buffer size values for coordinator nodes, we may improve the message throughput in cluster-tree WSNs, we may reduce problems related to congestion, high end-to-end delays and discarded messages due to buffer overflows. Most of these problems are directly related to inappropriate configurations of network communication parameters [55, 56, 64, 65].

1.4 Main Contribution and Outline

The main research contributions and the remainder of this Thesis are organised as follows.

Chapter 2 presents an overview of the state-of-the-art regarding routing protocols for WSNs, focusing on their adaptation to wide-scale applications and the impact that WSN topologies have upon their real-time properties. This overview presents a set of existing works, that may be of relevance for the understanding of the proposed approaches in this Thesis. The most relevant part of this chapter was published in [66].

Chapter 3 presents a set of modular simulation models for IEEE 802.15.4/ZigBee networks, that are able to deal with wide-scale cluster-tree WSNs and to address some of their main research issues. The proposed set of simulation models was developed according to the standards and provides important mechanisms to deploy and test wide-scale cluster-tree networks and their data communication mechanisms, allowing to easily design and test new implementations. This chapter is under consideration for publication in International Journal of Distributed Sensor Networks (IJDSN), SAGE.

In particular, this chapter presents the following contributions:

- A set of modular simulation models of IEEE 802.15.4/ZigBee cluster-tree WSNs to implement random or customizable wide-scale deployments;
- Implementation of the main Cluster-tree network mechanisms, such as: cluster-tree formation considering several network attributes, hierarchical addressing, beacon scheduling, and static and proportional superframe duration allocation schemes;
- Implementation of direct and indirect data communication mechanisms;
- Customizable data message streams modelling, considering both monitoring data traffic (node to sink communication) and source-to-destination data traffic (any source to any destination communication).

Chapter 4 presents a set of proportional allocation schemes that enable the setting-up of the superframe duration, beacon interval and buffer size values for cluster-heads of IEEE 802.15.4/ZigBee cluster-tree networks, considering the timing constraints imposed by the set of message streams generated by sensor nodes. This chapter is under consideration for publication in Sensors, MDPI.

In particular, this chapter presents the following contributions:

- The formulation of a set of worst-case boundary equations for IEEE 802.15.4 cluster-tree networks, defining the constraints that must be fulfilled during the network operation, in order to guarantee the adequate timing behaviour of the supported applications.
- The definition of a set of guidelines based on the traffic requirements and network topology to define the superframe durations, beacon intervals and the buffer sizes for each cluster-head, in order to significantly improve the network throughput, avoiding or reducing the congestion of the cluster-tree network.

Chapter 5 presents a novel alternative communication approach, that enables the support of peer-to-peer message streams in IEEE 802.15.4/ZigBee cluster-tree networks, with guaranteed QoS requirements. A simulation assessment of the proposed alternative communication scheme is also presented. This chapter is under consideration for publication in Sensors, MDPI.

In particular, this chapter presents the following contributions:

- The definition of a new alternative communication scheme for cluster-tree WSNs, called *Alternative-Route Definition* (ARound) communication scheme, that avoids the use of default cluster-tree paths; instead, it defines alternative communication paths between source and destination nodes, by using border nodes to relay messages. This scheme saves resources of the cluster-tree network, allowing to extend the cluster-tree network lifetime, because it saves energy from cluster-head nodes, which have a crucial role in keeping the network backbone working.
- The behaviour of the ARound scheme has been assessed and verified through simulation, highlighting its advantages in what concerns the reduction of both end-to-end communication delays and energy consumption of the overall network.

Finally, **Chapter 6** presents the conclusions of this Thesis and provides some new research directions.

A complete list of published/submitted papers is given at the end of this document, in Appendix A.

This page was intentionally left in blank

Supporting Real-Time Communication in Wide-Scale Wireless Sensor Networks

This chapter presents an overview of the state-of-the-art regarding routing protocols for WSNs, focusing on their adaptation to wide-scale environments and the impact that topologies have upon their real-time properties. This chapter is an adaptation of the following book chapter:

- LEÃO, Erico Meneses; VASQUES, Francisco; PORTUGAL, Paulo. Supporting Real-Time Communication in Large-Scale Wireless Sensor Networks. In: *Encyclopedia of Information Science and Technology*. IGI Global, Hershey, PA, USA, Third Edition, 2014, pp. 7371-7380.

2.1 Introduction

Automation activities are essential for the competitiveness increase in all industrial sectors. According to Chen et al. [67], an increasing number of industrial applications are focusing on wireless networks as a core technology. Wireless technologies have also been identified as a very attractive option for industrial and factory automation, distributed control systems, automotive systems and other kinds of networked embedded systems [2]. Within this context, wireless sensor systems can revolutionize industrial processing and meet the demand for increased competitiveness. Wireless sensor technology offers reliable and autonomous communication services for process control, in order to improve product quality, increasing yields and reducing costs. Thus, the use of Wireless Sensor Networks (WSN) in the industrial sectors is an attractive topic of research. Multiple research works have addressed routing and energy-efficiency problems. However, little attention has been given to the support of real-time communication in wide-scale WSNs.

The WSN literature proposes two research contexts: standards and routing protocols. In the standardisation context, one of the most frequently used WSN technologies is the IEEE 802.15.4 standard. The IEEE 802.15.4-2011 [68] is a standard designed for Low-Rate Wireless Personal Area Networks (LR-WPANs), which focus on short-range operation, low-data rate, energy-efficiency, and low-cost implementations. However, this standard has a number of limitations that prevents the construction of wide-scale wireless sensor networks with temporal constraints. Within this context, the IEEE 802.15.4e [19] and 802.15.5 [20] standards were released to enhance the timing/timeliness properties of the 802.15.4 MAC protocol (e.g. lower latency and higher robustness and determinism), and for increasing the

network scalability through the use of mesh topologies. The IEEE 802.15.4e standard introduces additional medium access control (MAC) mechanisms and frame formats that allow a wide range of industrial and commercial applications. However, this standard also has a set of limitations, as the reduced number of performance evaluation works in wide-scale networks [69]. The IEEE 802.15.5 standard proposes a mesh topology, by implementing a logical tree and mesh links, increasing the network scalability.

In the routing protocol context, the literature addresses several challenges and special considerations in the design of routing protocols for WSNs [30, 60, 63, 70]. Multiple research works have been focusing on routing and energy-efficiency problems. However, due to the real-time requirements imposed by automation applications, new approaches and algorithms must be provided to wide-scale WSNs.

This work discusses the state-of-the-art of standards and routing protocols for WSNs, focusing on their adaptation for wide-scale WSNs and the impact that WSN topologies have upon their real-time properties. Firstly, a background and discussion about the standards for WSNs and their main features are provided, encompassing real-time and wide-scale aspects. Then, within this context, several research works about routing protocol in WSN are presented. Finally, this work presents the conclusions and final considerations.

2.2 Background

Wireless Sensor Networks (WSNs) are special *ad hoc* networks that can be used in several application areas, such as: home, health, and environmental monitoring, robotic, vehicle systems, and military applications. With the advent of new technologies in the context of wireless communication, digital microelectronics, it is possible to build wireless sensor devices with increasing robustness, and more storage, processing, and energy capabilities. Within this context, with these capabilities and reduction of device size and costs, it becomes possible the design of wide-scale WSNs.

Over the years, several standards have been defined for wireless sensor networks. The IEEE 802.15.4 standard is one of the most commonly used wireless Sensor Network technologies. The IEEE 802.15.4-2011 standard defines the PHYsical layer (PHY) and Medium Access Control (MAC) sublayers specifications for Low-Rate Wireless Personal Area Network (LR-WPAN). Low-Rate Wireless Personal Area Networks are a simple, low-cost communication networks that allow wireless connectivity in applications with limited power and relaxed throughput requirements [68].

However, the IEEE 802.15.4 standard has a number of limitations that prevent the design of wide-scale wireless sensor networks with temporal constraints, such as: short-range communications with a small bandwidth, not adequately addressing critical requirements and determinism, only seven Guaranteed Time Slots (GTS), single channel, and lack of multihop communications and mesh capabilities. In this context, the IEEE 802.15.4e and IEEE 802.15.5 standards were released, in order to minimize the limitations of the IEEE 802.15.4 standard. On the one hand, the IEEE 802.15.4e standard implements a number of improvements, in order to support the industrial market requirements, such as: a new multi superframe structure, extension of the GTS slots and frequency channels (channel diversity). However, this standard also has a set of limitations, for instance, in wide-scale deployments,

topology changes can generate additional overheads due to the rescheduling of beacons, which increases the energy consumption. Moreover, whenever information can not be dispatched in a unique superframe, it is necessary to wait the next multi superframe, which can generate additional delays. In addition, there is a lack of performance evaluation works for wide-scale networks [69].

On the other hand, the IEEE 802.15.5 standard proposes a mesh topology through the implementation of a logical tree and mesh links, increasing the network scalability. Nevertheless, the main problems of the IEEE 802.15.5 standard are the lack of a complete performance evaluation [69], besides it does not provide end-to-end reliability and security [20].

A series of relevant works were proposed in literature, addressing different approaches, such as: performance evaluation, solutions and reviews of the IEEE 802.15.4 standard [52, 71–73]. Chen et al. [67], Jeong and Lee [74], and Stanislawski et al. [75] have presented solutions and performance evaluation of the IEEE 802.15.4e. Moreover, Lee et al. [76], and Cho et al. [77] have presented solutions and reviews of the IEEE 802.15.5 standard.

2.3 Challenges of Real-Time Communication in IEEE 802.15.X

Due to the typical requirements of WSN applications, which may impose a large number of sensor nodes deployed in a specific environment with the purpose of sensing different physical variables, there are a large number of WSN open research questions. In fact, there are several works in the literature for IEEE 802.15.4-based wireless sensor networks, addressing different issues. However, most of these works focus on energy consumption and efficiency.

According to Chen et al. [73], the real-time aspects are not a primary concern. Also, Teng and Kim [78] point out the energy-efficiency issue as the main question of most works in WSNs and they highlight some research challenges for designing real-time communications in WSNs. In this context, the main target of conventional WSN protocols is to maximise the average response time of the overall network. In real-time WSNs, the main target is to ensure the individual message deadlines.

Regarding real-time issues, several WSN applications need to operate in real-time, i.e., data packet transmissions have time constraints. For example, fire-fighting applications [11], where appropriate actions should be immediately performed in the monitored area; accelerometer-based speed estimation, in order to avoid vehicular crashes [17]; military operation applications [78]; and a wide range of industrial applications. The design of real-time protocols for WSNs becomes many challenger due to the large number of constraints that must simultaneously be satisfied.

Stankovic et al. [17] have introduced challenges for real-time communications in WSNs, addressing important questions, in which it is important to highlight:

- General research challenges: constraints that must simultaneously be satisfied, such as: paradigm shift, resource constraints, unpredictability, high density/scale, real-time, and security.

- Networking research challenges: the main challenges in the MAC, network and transport layers, providing a comprehensive view of the real-time operation in these layers.

With this, the authors highlight the need of designing new network protocols to deal with specific WSN requirements, mainly regarding to real-time, since the protocols of conventional *ad hoc* networks are not suitable for WSNs.

Also, Channa and Memon [79] highlighted the challenges regarding real-time communication in WSNs. For the authors, real-time applications impose several guarantees in terms of throughput and end-to-end delays. For this, several important considerations must be taken into account for the design of network protocols, in order to ensure acceptable Quality of Service (QoS) for WSNs. For instance, WSNs must guarantee bandwidth for real-time and best-effort traffic, which can be obtained through adequate multipath protocols. Besides, other relevant questions can be also highlighted, such as: removing of data redundancy through data aggregation, trade-off between energy consumption and delays, and support of multiple traffic types. The authors point out that few works were performed in order to combine techniques for real-time scheduling and energy-efficiency issues.

Recently, Teng and Kim [78] have presented a survey on real-time MAC protocols for WSNs. In general, the presented protocols were designed to ensure timing constraints for real-time applications. However, they usually fail in crucial issues such as: network synchronization, random deployment, and, in some cases, loss of deadlines under high congestion conditions. Also, several weaknesses in the MAC protocols were identified regarding to network scalability. Moreover, the authors emphasize that most of the designed protocols only decrease the packet transmission latency from source to destination, and they do not consider a message scheduling algorithm based on message deadlines. In other words, most of real-time algorithms provide low latency, but does not provide guarantees for real-time applications. Also, an acceptable trade-off between energy conservation and latency must be ensured, besides minimising the control overhead, which is a great challenge.

Chen et al. [73] have presented a relevant work regarding IEEE 802.15.4-based real-time communication. In this work, the authors show some of the main limitations of IEEE 802.15.4 standard, that prevent its applicability for delay bounded real-time applications. For this study, the authors consider a typical industrial automation application, performing analytical and simulation experiments, in order to verify the capability of the standard to meet specific real-time requirements. According to the analytical protocol evaluation, they showed that the conventional IEEE 802.15.4 standard does not meet the requirements of guaranteed latency, even considering a simulation environment with only 7 device nodes (maximum GTS slots of the standard).

The main limitations of the IEEE 802.15.4 standard to guarantee real-time capabilities are related to the limitation of the number of GTS slots and the initial configuration of the CSMA-CA parameters. The IEEE 802.15.4 standard only defines a maximum number of seven GTS slots, which means that the number of devices involved in this communication type is strictly limited. Regarding to the CSMA-CA parameter configuration, the standard values also restricts communication guarantees [73].

Chen et al. [73] propose a protocol structure modification for the IEEE 802.15.4 standard. This new protocol version was designed for real-time industrial applications and represented

a basis for the standardization of the IEEE 802.15.4e standard. The authors have considered 20 sensor nodes in a star topology, a guaranteed upper latency boundary of 10 ms, and the PHY layer is totally preserved, in order to maintain hardware compatibility. The superframe structure is based on TDMA scheme, where the Contention Access Period (CAP) was completely removed. Thus, the new superframe structure is basically composed of GTS slots, in which sensor nodes transmit alarm messages to the PAN coordinator. GTS slots need to be preallocated to each of the devices [73]. This protocol provides two operation modes: beacon tracking enabled or disable. In enabled mode, PAN coordinator sends periodically beacon frames in order to maintain synchronisation of all nodes. In disable mode, after receiving the first beacon frame, devices can immediately sleep to save energy. To transmit a new message, devices need to resynchronise with the PAN coordinator by tracking the next beacon frame.

According to authors, using beacon tracking enabled mode, the guaranteed latencies are respected. Unlike, using beacon tracking disable mode, the maximum guaranteed latency is not respected. Although these results point out a poor performance of the beacon tracking disable mode, this mode can be applied for more flexible scenarios with higher latencies. Note that the proposed scenario is very limited regarding to scalability issues. Although providing guaranteed latencies for a typical industrial application, this protocol modification can not be applied to wide-scale applications, which is the focus of this research work. However, these results showed that the change of IEEE 802.15.4 structure parameters may present adequate results for applications with timing requirements.

Xia et al. [80] have also presented a survey of adaptive and real-time protocols based on the IEEE 802.15.4 standard. They discuss some of the negative aspects of the IEEE 802.15.4 MAC protocol (CAP and CFP), related to adaptive and real-time requirements. Furthermore, they presented a review of mechanisms used by adaptive and IEEE 802.15.4-based real-time protocols. According to the authors, the slotted CSMA-CA algorithm does not provide service differentiation mechanisms for time-critical events, such as alarm messages. Moreover, the initial values for the CSMA-CA parameters can influence the network performance. Therefore, based on the initial parameter configuration, the network performance must be assessed in order to identify unacceptable side effects. Although the authors present a series of research works that improve the network performance regarding to real-time requirements, more research studies are needed in order to solve some remaining problems, such as: high latencies, large power consumption, system complexity and implementation overhead.

2.4 Wide-Scale and Real-Time Routing Protocols in WSN

Basically, the main function of the routing protocols is to find and establish routes between two entities that wish to communicate. Thus, routing algorithms are an important research topic in computer network field. In the last few years, several routing protocols for *ad hoc* networks were proposed in the literature. However, routing protocols for *ad hoc* networks are typically unsuitable for WSNs [9].

Therefore, one of the most important topics in WSNs is the design of adequate routing protocols. WSNs impose a set of challenges that must be considered for the design of routing

protocols, such as: energy-limited nodes, dynamic networks, randomly deployment, data redundancy, lack of network backbone, time-constrained applications, and wide-scale deployment [30, 60, 63, 70].

Several works have presented comprehensive and extensive surveys of routing techniques in WSNs, emphasizing important topics, such as: lifetime mechanisms, energy efficiency issues, multihop approaches, clustering algorithms, wide-scale deployments, and real-time requirements. Each one presents a set of routing protocols, highlighting their strengths and weaknesses, based on different metrics and points of view. These surveys propose new taxonomies and classifications. In general, most of works classify routing protocols in three groups, based on their network structure: data-centric, hierarchical, and location-based schemes [30, 70, 81–83].

In data-centric or flat routing protocols, all sensor nodes perform the same functions and their activities are related to sensing tasks and network control. In this protocol class, flat algorithms implement query-based mechanisms, where sink nodes request data from sensor nodes of a given monitored area [70]. In location-based routing protocols, sensor nodes are addressed based on their locations. Thus, location information is used in order to transmit data for desired regions. Location information can be obtained via Global Positioning System (GPS) devices or by message exchanging between neighbour nodes [82, 83].

Unlike flat protocols, hierarchical protocols apply clustering mechanisms in order to generate clusters of sensor nodes, creating different responsibility levels and rules among the nodes. High-level nodes are responsible by network control and data aggregation activities, while the lower-level nodes perform only sensing activities. The protocol LEACH (Low-Energy Adaptive Clustering Hierarchy) [84] is one of most used and cited hierarchical protocols in the literature. Several hierarchical protocols were designed based on LEACH. LEACH is an energy-efficiency protocol for WSNs, in which sensor nodes continuously send data toward the sink node. Sensor nodes are grouped into local clusters and a specific node is chosen as the cluster-head (CH) or local base station (BS). Cluster nodes are responsible for sending data to their CHs, using CSMA-CA protocol. In turn, CH nodes are responsible for aggregating data of member nodes and sending it to the sink node, using Time Division Multiple Access (TDMA) communication.

LEACH includes a randomized rotation of the CH nodes, based on the highest-energy nodes. This mechanism is performed in order to save energy of the nodes and to increase the network lifetime. The LEACH algorithm considers that if a specific node is always the cluster-head, its energy is drained and this node would quickly die, decreasing the cluster lifetime. Thus, this algorithm evenly distributes the energy consumption for all nodes, which may maximise the network lifetime. Regarding to synchronisation issues, all sensor nodes initialise a duty cycle at the same time.

Ahmed and Fisal [11] proposed a real-time routing protocol with load distribution in WSNs (RTLTD). This protocol ensures high throughput with minimized packet overhead and prolongs the network lifetime through a forwarding decision based on link quality, packet delay and the remaining energy of next hop sensor nodes. RTLTD uses a communication scheme named Geodirectioncast Forwarding, which combines geocast with directional forwarding to relay data packets to destination, using multipath.

The RTLTD scheme combines four functional modules: location, routing, power, and neighbourhood managements. The location management defines information about the

localization of nodes based on the distance from the node itself to three-neighbour nodes. The routing management computes the optimal forwarding based on the packet speed, link quality and remaining energy. The power management adjusts the transceiver power and defines the level of transmission power for the sensor node. The neighbourhood management is used to discover neighbour-forwarding nodes and to maintain a neighbour table. Based on simulation and test bed, the authors showed that RTLD provides a high delivery ratio and good performance regarding to energy consumption and packet overhead.

Hanzalek and Jurcík [52] and Koubaa et al. [71] have presented relevant works regarding real-time issues. The key idea of these works is to define adequate clusters' active period scheduling, considering the main constraints imposed by data flows, guaranteeing the real-time requirements of the flows. Koubaa et al. [71] proposed a Time Division Beacon frame Scheduling (TDBS) mechanism, in order to avoid intra-cluster beacon collisions for IEEE 802.15.4 cluster-tree networks. The TDBS mechanism is based on a time division approach to build synchronised multihop cluster-tree WSNs. This mechanism considers the Superframe Durations and Beacon Intervals of a set of coordinator nodes in order to define a free-collision beacon frame scheduling. Hanzalek and Jurcík [52] have presented a Time-Division Cluster Scheduling (TDCS) mechanism, to schedule the clusters' active periods and to meet all end-to-end deadlines of time-bounded flows. For this, TDCS uses an integer linear programming algorithm to solve the scheduling problem.

Currently, the literature proposes the use of hierarchical routing protocols to solve the energy problem in wide-scale WSNs [30]. The use of hierarchical routing protocols presents some advantages for wide-scale application, when compared with flat approaches, such as: reduction of the size of routing tables, decreasing of communication bandwidth, distribution of the management tasks, data aggregation, reduction of contention of nodes, and stabilization of network topology at the level of sensors [14]. Within this context, several hierarchical routing protocols for WSN were proposed.

Abbasi and Younis [14], Boyinbode et al. [85], and Liu [60] presented surveys on clustering algorithms for WSNs. Abbasi and Younis [14] presented a taxonomy and general classification for clustering algorithms, based on the cluster properties, CH capabilities and the clustering process. Moreover, the authors compared a set of clustering protocols using different metrics, presenting their main advantages and weaknesses. Boyinbode et al. [85] summarised existing clustering protocols and highlighted the main challenges for clustering, such as: cluster formation mechanisms, cluster-head selection, limited energy, network lifetime, synchronisation, data aggregation, fault tolerance, and others. In turn, Liu [60] have presented an extensive survey on clustering routing protocols for WSNs and proposed a novel taxonomy based on a set of clustering attributes, such as: cluster characteristics, cluster-head characteristics, and clustering process.

Although the above surveys cover most of the known routing protocols for WSNs, no one takes into account the requirements imposed by wide-scale deployments. In this context, Li et al. [30] have presented a survey on routing protocols for wide-scale WSNs. In this work, the authors highlighted the need of using hierarchical routing protocols for wide-scale deployments. They proposed a new taxonomy for hierarchical routing protocols for wide-scale WSNs, based on different categories, such as: control overhead reduction-based, energy consumption mitigation-based, and energy balance-based categories. In the control overhead reduction-based category, routing protocols use innovative approaches for

simplifying the routing process and reducing the control overhead. In the energy consumption mitigation-based category, the routing protocols aim to decrease the energy consumption, through the dynamic event clustering, multihop communication, and cooperative communication. In the energy balance-based category, the main objective of routing protocols is to balance the energy consumption for the nodes, by equally distribute the network activities for all nodes. Moreover, the authors summarise the main routing protocols for wide-scale WSNs.

2.5 Future Research Directions

Recently, several IEEE 802.15.4-based research works have been proposed in the literature. However, most part of these research works address routing and energy-efficiency problems and little attention has been given to real-time communication in wide-scale WSNs. Thus, real-time and wide-scale issues are a very attractive research topics in WSNs.

Based on this study, we conclude that a combination of techniques seems to be promising, when designing approaches to support real-time communication in energy-efficient and wide-scale WSNs. Regarding to scalability issues, the use of hierarchical algorithms can be pointed out as a crucial question for the design of communication protocols for wide-scale WSNs, due to several features presented along this chapter. In the context of real-time, the timing requirements imposed by messages must be guaranteed. This way, it is important the use of efficient approaches that ensure bandwidth for the message streams, such as: adequate allocation schemes for the communication structures, reduction of control traffic, and reduction of congestion. Furthermore, the design of new alternative communication schemes is an attractive research topic, in order to avoid problems of traditional hierarchical communication approaches, such as: high congestion and end-to-end delays.

2.6 Conclusion

In general, there are several research works about WSNs involving multiple issues such as: standardization studies, energy-efficiency, real-time and wide-scale routing protocols, and MAC protocols. However, due to specific requirements imposed by WSN applications, the design of energy-efficient and real-time communication protocols for wide-scale WSNs is even more challenging. Most the proposed research works only consider limited environments, and scalability issues are not satisfactorily addressed. Thus, the design of new energy-efficient schemes to deal with real-time and wide-scale issues is an attractive and challenger research topic.

CT-SIM: A Simulation Model for Wide-Scale Cluster-Tree Networks based on the IEEE 802.15.4 and ZigBee Standards

This chapter presents an IEEE 802.15.4/ZigBee-based simulation model developed for the OMNeT++/Castalia Network Simulator, that enables the simulation assessment of wide-scale deployments of cluster-tree WSNs. This chapter is a reproduction of the following manuscript:

LEÃO, Erico; MORAES, Ricardo; MONTEZ, Carlos; PORTUGAL, Paulo; VASQUES, Francisco. CT-SIM: A Simulation Model for Wide-Scale Cluster-Tree Networks based on the IEEE 802.15.4 and ZigBee Standards. **SUBMITTED TO:** *International Journal of Distributed Sensor Networks*, SAGE, 2016.

3.1 Abstract

The IEEE 802.15.4/ZigBee set of standards is one of the most used Wireless Sensor Network technologies. These standards support cluster-tree networks, which are suitable topologies for wide-scale deployments. The design of wide-scale wireless sensor networks is a challenging task, because it is difficult to test, analyse and validate new designs in real scenarios. Thus, simulation becomes a convenient and feasible method for its assessment before deployment. Within this context, we propose a new simulation model for IEEE 802.15.4/ZigBee-based networks, that is able to deal with wide-scale cluster-tree WSNs and to address its major challenges. The proposed simulation model provides important mechanisms for the assessment of wide-scale cluster-tree networks and associated data communication mechanisms, enabling an easier design and test of wide-scale WSN implementations.

3.2 Introduction

Wireless Sensor Networks (WSN) are special *ad hoc* networks composed of hundreds of sensing devices that can be used in several application domains, such as: home, health and environmental monitoring, robotics, industrial automation, vehicle systems, and military applications [86].

The IEEE 802.15.4/ZigBee set of standards is one of the most used WSN technologies. The IEEE 802.15.4 standard [18] defines the PHYsical layer (PHY) and Medium Access Control (MAC) sublayers specifications for Low-Rate Wireless Personal Area Networks (LR-WPAN). The ZigBee standard [21] defines application and network layers for the IEEE 802.15.4 protocol stack. This set of standards defines the basis to build cluster-tree networks, which are suitable topologies to interconnect wide-scale deployments of wireless sensor devices.

Due to the difficulties associated to the deployment of real wide-scale WSNs, simulation approaches are commonly used by researchers and developers to test, analyse and validate algorithms and protocols during the design and development stages [87, 88]. In fact, testing new WSN approaches in real wide-scale environments can be a complex and time-consuming task, demanding huge efforts, and costs. This way, simulation tools are used to model parts of real world [89], reducing time and costs associated to the early stage assessment of the network.

Although the availability of multiple simulation tools, most of these tools are adequate to assess just networks with star topologies, as more complex topologies such as cluster-tree are usually not addressed.

In this paper¹, we present *CT-SIM*, a simulation model for IEEE 802.15.4/ZigBee-based WSNs. The proposed simulation model is able to deal with cluster-tree topologies configured randomly or in customized way, as it implements the most relevant formation and communication mechanisms that are required to build-up these type of networks.

The main contributions of CT-SIM simulation model can be summarized as follows: (1) random or customizable node deployments under wide-scale environments, (2) cluster-tree formation mechanisms considering attributes such as the maximum number of nodes per cluster and the maximum depth for the tree, (3) hierarchical addressing, (4) customizable beacon scheduling mechanisms, in order to avoid interferences among clusters, (5) static and proportional superframe duration allocation schemes, (6) direct and indirect data communication mechanisms, and (7) customizable data message streams modelling, implementing both monitoring data traffic (node to sink communication) and source-to-destination data traffic (any source to any destination).

The proposed CT-SIM simulation model was developed upon the Castalia Simulator [90], that provides a modular structure allowing the implementation of new communication issues. To the best of our knowledge, no other available simulation model addresses the above listed communication mechanisms, allowing complex simulations of cluster-tree WSNs in random environments. One of its main advantages is that most of its functionality are fully integrated, which saves time and complexity, when compared with manual interactions among simulation models (option provided by most simulators). A beta version of the CT-SIM simulation model will be available soon, as part of the Castalia database.

3.3 Overview of the IEEE 802.15.4/ZigBee set of standards

The IEEE 802.15.4 standard [18] defines the PHYsical layer and the Medium Access Control (MAC) sublayer for Low-Rate Wireless Personal Area Networks (LR-WPAN), focusing on

¹The term "*paper*" is commonly used along this thesis, in order to keep the full compatibility with the content of the referred manuscripts.

short-range operation, low-data rate, energy-efficiency and low-cost implementations.

Basically, this standard supports two network topologies: star and peer-to-peer. In star topology, sensor nodes directly communicate with a central node (the PAN coordinator), which is responsible to initiate, terminate and route all communications in the network. The main weakness of a star topology is that the network coverage is limited by the transmission range of its member nodes, which prevents its usage in wide-scale deployments.

A peer-to-peer topology allows to implement more complex network formations, such as mesh and cluster-tree topologies. The latter enables the implementation of wide-scale network deployments, through a mesh of multiple neighboring clusters. The cluster-tree network topology is detailed in the next subsection.

3.3.1 Cluster-tree network

Although suggesting the use of cluster-tree topologies, the IEEE 802.15.4 standard does not define the required mechanisms to deal with this type of topologies. The ZigBee standard [21] can be assumed as a complementary standard because it provides the communication services and protocols that allow the building-up of cluster-tree topologies. Basically, in a cluster-tree topology, nodes are grouped into clusters. A cluster-tree network is formed by cluster coordinators (named cluster-heads) that provide synchronization to its member child nodes, which can be end nodes or other coordinators. A coordinator node can connect to a new coordinator and form a multicluster hierarchical network structure. Figure 3.1 illustrates the implementation of cluster-head nodes in an IEEE 802.15.4 cluster-tree wireless sensor network.

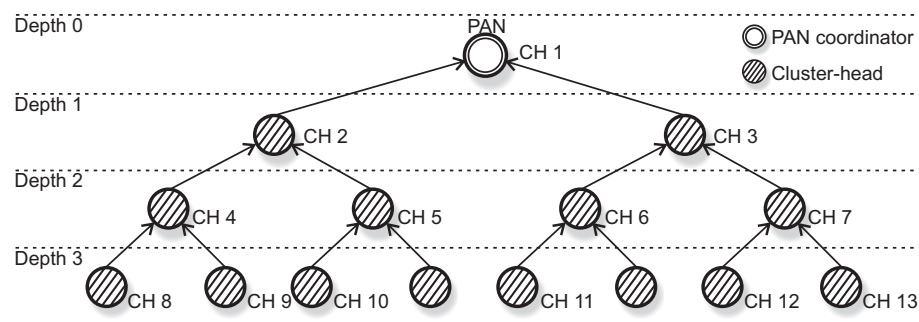


Figure 3.1: IEEE 802.15.4/ZigBee cluster-tree network.

When using cluster-tree topologies, the network operates in beacon-enabled mode. In this mode, message exchanges are organized according to a structure called superframe. A superframe is bounded by beacons, which are periodically transmitted by cluster-heads. These beacons synchronize the cluster nodes, identify the PAN and describe the superframe structure. A superframe structure is formed by two parts: active and inactive periods. In the inactive period, nodes can enter in low power mode to save energy. The active part comprises two periods: contention access period (CAP) and contention-free period (CFP). During the CAP, a node device wishing to communicate competes with other devices using a slotted CSMA-CA mechanism. The CFP period is used for applications that require low latency or specific bandwidth. During this period, the active part is dedicated to the device allocated in the referred CFP, in GTS slots (Guaranteed Time Slots).

The superframe structure is configured by two parameters: *macBeaconOrder* (BO) and *macSuperframeOrder* (SO). These parameters define the beacon interval (BI) and the superframe duration (SD), respectively. The BI determines the interval at which the coordinator shall transmit its beacon frames. The SD determines the length of the active portion of the superframe. Figure 3.2 illustrates the superframe structure and describes the SD and BI values.

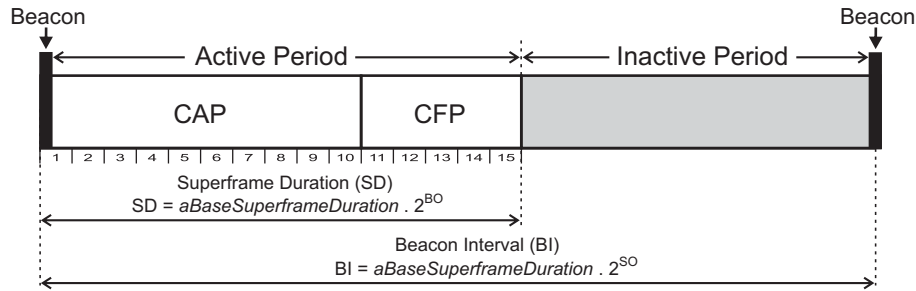


Figure 3.2: IEEE 802.15.4 Superframe structure

3.4 Related work

The availability of a wide range of WSN simulators has been the focus of some recent survey works [87, 89, 91–93]. In general, these surveys address state-of-the-art simulation tools and frameworks for wireless sensor networks, comparing their main features, strengths and weaknesses.

Živković et al. [87] present an exhaustive survey of state-of-the-art simulators for wireless sensor networks. According to the authors, the main objective of this survey is to help different groups of users to select appropriate simulators for different requirements. For this, the authors classify the simulators in three domains of use: education, research, and industrial. Another exhaustive survey was presented by Dwivedi [89]. In this work, the authors review several simulation tools based on different perspectives, such as: simulation frameworks, emulators, visualization tools, test-beds, debugging tools, code-updating/reprogramming tools, and network monitors.

Korkalainen et al. [91] provide a survey addressing five simulation tools (NS-2, OMNeT++/Castalia, Prowler, TOSSIM and OPNET) commonly used in the literature. The authors review these simulators based on attributes such as: real-time task modelling, energy models, mobile sensor support, radio models, routing protocols, scalability for wide-scale network, among others.

Feeney [92] presents a survey addressing only OMNeT++-based simulation tools that support IEEE 802.11 and IEEE 802.15.4 standards. In this work, the following simulators are reviewed: Inet, Inetmanet, MiXiM, Mobility-fw, and Castalia. The author proposes a well-defined scenario framework in order to evaluate the behaviour of simulation components and to perform simulation experiments using each simulator. With this, the author demonstrates the need of adopting common tests and scenarios in order to increase the reliability of the simulator assessments.

Table 3.1: Comparison of the most used WSN simulators.

	NS	TOSSIM	Prowler	OPNET	OMNeT++	Castalia
Type	Simulator	Emulator	Simulator	Simulator	Simulator	Simulator
GUI support	Poor	YES	YES	YES	YES	YES
Energy consumption model	YES	NO	NO	YES	YES	YES
IEEE 802.15.4 support	YES	Basic CSMA	Poor	YES	YES	YES
Scalability	YES	YES	NO	YES	YES	YES
Clustering	YES	Not specified	Not specified	YES	YES	YES
Wide-scale support	Limited	YES	YES	YES	YES	YES
Mobility	YES	NO	YES	YES	YES	YES
Strengths	Extensions; Available models; Clock drift; Battery model; Channel model; Protocol stack;	Hidden nodes	MATLAB tools; Radio models; QoS metrics;	Realistic radio; Custom packet format; Available models;	Channel model; Clock drift;	Realistic radio; Wireless channel; Clock drift; MAC, Routing and App layers; physical process model; Allow extensions;
Weaknesses	High simulation time; Complex operations; Low support for WSN MACs;	Only support MICAz; Same code for all nodes;	Only TinyOS MAC;	Commercial; High complexity;	Lack of simulation models;	Lack of simulation models;

Khan et al. [93] present a survey of simulation tools for wide-scale WSNs, using several comparison criteria, such as: popularity, accessibility, complexity, accuracy, extensibility, scalability, and availability of models and protocols. The authors conclude that there is no an *all-in-one* simulator for wireless sensor networks. In fact, available simulation tools provide own features and models for specific applications. Other important analysis is that different simulation tools present different results for the same simulation model. This is due to the influence of the used resources and features and, mainly, due to the simulation configuration and communication scenario assumptions.

Moreover, Stojmenovic [94] provide important advices about simulation practices. The authors dictate that simulation assessments should use simpler models, with well-defined assumptions and metrics, and their complexities can gradually be increased, adopting new assumptions, metrics and environment for the algorithms. The authors advocate the idea that new proposed protocols, conceptions or theoretical analyses can use simulations in order to provide initial support, identifying possible weakness that can be addressed by further researches.

Table 3.1 summarises available simulators proposed in literature, considering the most relevant requirements within the scope of this paper, that is, their suitability to support wide-scale implementations of IEEE 802.15.4 WSNs.

Network Simulator (NS) is one of the most used network simulators. It is a free and open general purpose discrete event simulation tool, widely used by the community. The

current NS-3² version is totally written in C++ language with possible Python bindings. NS-3 can be used by researchers to test and analyse their protocols and wide-scale systems in a controlled environment [87]. Network Simulator provides support for simulation of Internet-based protocols and wireless networks, with several extensions and models. As strengths, we can highlight: support to IEEE 802.15.4 MAC, support to mobility, allow extensions, flexibility, a large user community, graphical support, and battery modelling. However, the main drawback is its poor scalability, mainly regarding to the simulation time, which is undesirable for the simulation of wide-scale or dense WSNs.

TOSSIM [95] is a MICA Mote simulator for TinyOS³, which is an operational system designed for embedded WSNs [93]. As TOSSIM can run Mica mote code, it is considered an emulator, allowing to map hardware interrupts into discrete events [87]. According to Korkalainen et al. [91], TOSSIM was developed based on four key requirements: scalability (simulator must handle a large of number of devices and configurations), completeness (simulator must capture behaviour and interactions of a network at a wide range of levels), fidelity (regarding to capture an accurate behaviour of the network), and bridging (simulator must bridge the gap between algorithm and implementation). As strength, being TOSSIM an emulator, it simulates both the hardware and software of sensor devices. As weaknesses, TOSSIM does not provide energy consumption modelling and does not simulate the physical phenomena [89]. Besides, TOSSIM only emulates Mica Mote devices.

Prowler⁴ is a MATLAB-based discrete event simulator designed to simulate MICA mote devices. This simulator supports radio propagation models and collision detection models. However, as the TOSSIM simulator, Prowler was only designed to MICA mote devices and TinyOS MAC protocol [88].

OPNET⁵ is one of the most used commercial discrete event simulators (currently, OPNET modeller was integrated to Riverbed Modeller). OPNET (Riverbed Modeller) is a discrete event simulator for designing communication networks, allowing to develop models for different network entities. As a commercial solution, this modeller offers a large number of protocols and solutions, including the main wired and wireless communication networks. However, OPNET has a complex structure, which discourages its usage for academic domain [87]. Moreover, the high cost of commercial licenses (available free educational licenses) can also be considered as a weakness.

OMNeT++⁶ is an open-source discrete event network simulator based on components or modules. OMNeT++ uses C++ language to write components for the models and uses a high-level language (NED - Network Description Language) to assemble them. In general, OMNeT++ provides event scheduling, a message passing engine for its components, and languages to write and model simulations. As strengths, this simulator provides an excellent graphic interface, allowing tracing and debugging, and an excellent scalability and support for wide-scale networks. The main drawback is the lack of available protocols and energy modelling for wireless sensor networks [87, 91]. Several WSN simulation frameworks based on the OMNeT++ have been provided. Within the context of this paper, we focus on the Castalia simulator, which is a specific simulator for wireless sensor networks.

²The Network Simulator (NS-3): <https://www.nsnam.org>.

³TinyOS: <http://www.tinyos.net>.

⁴Prowler Simulator: <http://www.isis.vanderbilt.edu/projects/nest/prowler>.

⁵OPNET Modeller: <http://www.riverbed.com/gb/products/steelcentral/opnet.html?redirect=opnet>.

⁶The OMNeT++ Simulator: <https://omnetpp.org>.

Table 3.2: The main advantages of the proposed simulation model, compared with state-of-the-art simulation tools.

Comparison Topics	Network Simulator	Prowler	OPNET	OMNeT++	Castalia	CT-SIM
Cluster-tree WSN model	YES	Not specified	YES	With extensions	Only one-level	YES
Random network topology formation	YES	NO	YES	YES	YES	YES
Automatic beacon scheduling	NO	NO	YES	NO	NO	YES
Automatic active period allocation	NO	NO	YES	NO	NO	YES
Proportional Superframe Duration allocation	Not official	NO	YES	NO	NO	YES
Multiple data traffic mechanisms	YES	Not specified	YES	YES	YES	YES
Realistic Models	YES	YES	YES	YES	YES	YES
Scalability	Poor	Good	Good	Good	Good	Good

Castalia⁷ is an open-source discrete event simulator for WSNs, Body Area Networks (BANs) and general low-power embedded networks. It was developed at National ICT Australia (NICTA). This simulator is based on the OMNeT++ platform and it is widely used by researchers and developers to test their protocols using realistic wireless channel and radio models. Castalia implements an advanced wireless channel model based on empirically measured data. Also, the simulator provides radio models based on real low-power communication radios. Moreover, important features to simulate wireless sensor networks are available, such as: realistic node behaviour, node clock drift, and energy consumption model. Castalia is adaptive and expansible and was written in C++ programming language.

Castalia simulator provides a basic IEEE 802.15.4 model. However, this model is quite limited, as it only implements the CSMA-CA functionality and beacon-enabled PAN (star topology), including an association procedure, direct data transfer mode, and GTS communication. Hurtado-López et al. [96] provide a simple simulation model to support cluster-tree networks in the OMNeT++ simulator. This model is based on a previous one proposed by Chen and Dressler [97]. However, both models are limited and not scalable.

Within this context, the main target of the CT-SIM simulation model is to provide a simulation framework able to deal with wide-scale cluster-tree networks, encompassing their key features: random deployments, modular network formation, hierarchical addressing, cluster scheduling, and upstream and downstream traffic communication. Table 3.2 summarises the main advantages of the proposed CT-SIM simulation model compared to state-of-the-art simulation tools (included its official extensions and models) regarding to cluster-tree networks. For this comparison, we have excluded the TOSSIM tool, due to limitations imposed as emulator and its unique application (only MICAz in TinyOs). The proposed CT-SIM simulation model is presented in the next section.

3.5 CT-SIM WSN simulation model

In this paper, we propose CT-SIM, a new simulation model for IEEE 802.15.4/ZigBee-based cluster-tree wireless sensor networks. This simulation model was developed using the

⁷The Castalia Simulator: <https://castalia.forge.nicta.com.au>.

Castalia simulator. With CT-SIM, we provide an adequate communication environment to test algorithms and protocols for wide-scale cluster-tree networks, such as those presented by Leão et al. [98] and Felske et al. [99]. Importantly, the proposed model is completely adaptable and expandable.

In the proposed model, most part of the simulation parameters and network configuration schemes are defined in the *omnetpp.ini* (parameter configuration file). The simulation model itself is composed of three main phases: network formation, communication and statistic collection phases. Figure 3.3 highlights the implementation scheme of CT-SIM.

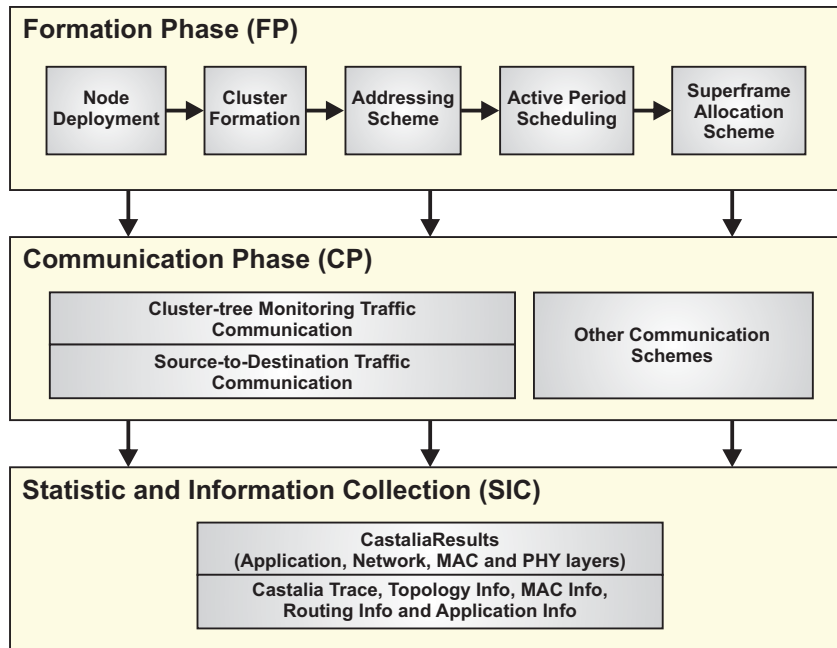


Figure 3.3: CT-SIM implementation scheme.

In the next subsections, we will present each phase and its main steps and features.

3.5.1 Formation phase

The first phase is the *Formation Phase* (FP), which is responsible for deploying the cluster-tree topology and to define the main mechanisms of the network operation. This is an important phase, as the network configuration and operation will depend of the methods and parameters used in this phase. FP is sequentially performed and its time duration is dependent of the number of sensing nodes. In general, this phase has a short time duration.

The *Formation phase* is composed of a set of steps, such as: node deployment, cluster formation, addressing scheme, active period scheduling and superframe duration scheme. Each step is detailed in the following topics.

3.5.1.1 Node deployment.

Basically, this step is responsible for defining the size of the monitored environment, the number of sensor nodes and their deployments. The number of sensor nodes is defined by

the simulation parameter `numNodes`, which is a mandatory parameter to be included in the parameter configuration file.

For the deployment of nodes, CT-SIM uses the deployment schemes provided by Castalia, such as: random uniform distribution and grid. Also, we can use a mixed deployment by associating a set of nodes to a specific deployment type. Besides, any node can be statically deployed through the `xCoor` and `yCoor` parameters. In turn, the environment size is defined by `field_x` and `field_y` parameters, which defines the axis-x and axis-y respectively. Table 3.3 shows the most commonly used parameters in the node deployment step.

Table 3.3: Parameters for the node deployment step.

Parameter	Definition
<code>numNodes</code>	Number of sensing nodes
<code>deployment</code>	Type of deployment for the nodes
<code>xCoor</code>	Position axis-x for a specific node
<code>yCoor</code>	Position axis-y for a specific node
<code>field_x</code>	Size of axis-x of the environment
<code>field_y</code>	Size of axis-y of the environment

3.5.1.2 Cluster formation.

For this step, the PAN coordinator (defined as the *root node*, at depth 0 of the cluster-tree network) is responsible to trigger the network formation process. For this, it broadcasts a *Cluster Formation* frame indicating its *cluster ID* and basic information such as the tree depth. Upon receiving this formation request frame, a joining node sends a *Cluster Association Request* frame, using the CSMA-CA mechanism to access the wireless channel. The PAN coordinator acknowledges all association requests through a *Cluster Association Response* frame. New association requests are accepted during a given time interval defined by *formationTime* parameter. Importantly, the cluster coordinators perform the *associationRequest_hub* function, which implements the admission control function for accepting (or not) new nodes. New association requests can be refused if the set of requirements cannot be fulfilled. Basically, for this version of the simulation model, the *associationRequest_hub* function bounds the maximum number of nodes for each cluster through the `maxChildren` parameter (by default, its value is set to 6 nodes). The Link Quality Indication (LQI) and Received Signal strength Indication (RSSI) values are also used as requirements to accept new nodes. Other admission control parameters can be included by implementing a different version for this function.

After accepting new associated nodes, the PAN coordinator performs the *CCH_Select* function. This function is used to select candidate nodes to be cluster-heads (CCH nodes). The main target of this function is to establish specific criteria to select the candidate nodes to build their own clusters, to increase the size of the cluster-tree network. For this, the maximum number of CH candidate nodes is defined through the `maxCCH` parameter (by default, its value is set to 3). Moreover, the RSSI value for nodes with an acceptable threshold is used in order to select the most distant candidate nodes. With this, we pretend to increase the maximum coverage of the cluster-tree, keeping a link quality between the parent coordinator and its child coordinators. Finally, the `maxDepth` parameter is defined aiming to limit the maximum

depth that the cluster-tree can reach. After reaching this specified depth, the CH selecting process is stopped, preventing the further growth of the cluster-tree. New selecting schemes of candidate nodes can be achieved by implementing different versions of the *CCH_Select* function. For example, CT-SIM may also provide alternative schemes based on the location of the nodes, in order to build more balanced cluster-tree networks.

Furthermore, each CCH node will perform the same cluster formation step, in order to deploy its own cluster, following the same rules. After finishing this process, the cluster-tree network is built. Table 3.4 shows the main parameters used in the cluster formation step.

Table 3.4: Parameters for the cluster formation step.

Parameter	Definition
<i>formationTime</i>	Time duration for association requests
<i>maxChildren</i>	Maximum number of child nodes
<i>maxCCH</i>	Maximum number of CCH nodes
<i>maxDepth</i>	Maximum depth of cluster-tree network

3.5.1.3 Addressing scheme.

CT-SIM defines unique network addresses for all sensor nodes. For this, it uses the ZigBee-based network hierarchical addressing scheme [21]. After the cluster-tree formation, each cluster-head requests its own address block, using a *bottom-up* approach. For this, cluster-heads send the *Addressing Request* frame to their parent coordinators, in order to request their address blocks based on the number of child nodes. After all the requests have reached the PAN coordinator, this node is the responsible for assigning the global address block, encompassing all network clusters. The address blocks are assigned among the cluster-heads using a *top-down* approach through the *Addressing Response* frame. Each cluster-head uses the first address of the address block for itself, and is responsible to assign the addresses for its leaf nodes and the respective address block for each child cluster-head. This process is performed along the tree, until reaching the last cluster-head. With this addressing approach, besides the address blocks of its child cluster-heads, each cluster-head keeps its own address block. New functionalities can be added to this addressing scheme by implementing alternative versions for this address requesting process. For example, an extra number of addresses may be added for each cluster-head in order to allow future associations.

3.5.1.4 Active period scheduling.

After the formation and addressing processes, the *cluster active period scheduling* is performed. The aim of this phase is to schedule the active periods of all clusters, in order to avoid interferences among overlapping clusters. In the CT-SIM simulation model, a time division scheduling scheme [26] was used through the *beaconSchedule* function. This function is performed by the PAN coordinator, which is responsible to schedule the cluster-tree network and to define the adequate *offset* for each cluster-head. This set of offsets corresponds to the start time during which each cluster must send its beacon frames.

In order to avoid inter-cluster collisions, an one-collision domain was implemented [51], i.e., only one cluster is active during a given time.

Basically, two cluster scheduling types were defined: *top-down* and *bottom-up* cluster scheduling schemes. In the *top-down* cluster scheduling scheme, the cluster active periods are ordered according to a *top-down* direction, starting with the PAN coordinator cluster. Then, clusters of the depth 1 are ordered, and later clusters of the following depths, so on, until reaching the maximum depth of the tree. Figure 3.4 illustrates the *top-down* cluster scheduling scheme for the cluster-tree network shown in Figure 3.1.

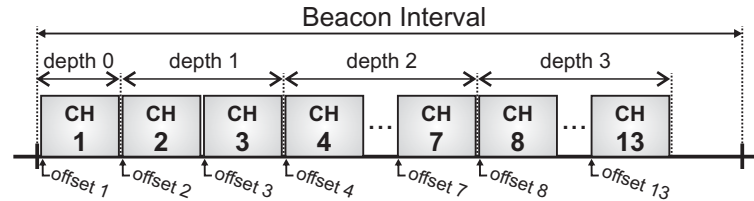


Figure 3.4: Top-down cluster scheduling scheme.

In the *bottom-up* cluster scheduling scheme, the cluster active periods are ordered in a bottom-up direction, i.e., firstly, the deepest clusters are ordered, then clusters of next lower depth, until the PAN coordinator cluster. Figure 3.5 illustrates the *bottom-up* cluster scheduling scheme for the cluster-tree network shown in Figure 3.1.

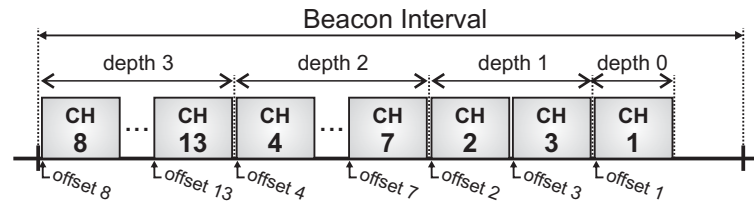


Figure 3.5: Bottom-up cluster scheduling scheme.

Note that the *bottom-up* scheduling scheme prioritises the upstream traffic, i.e., the traffic flowing from sensing nodes to the PAN coordinator (commonly the sink node of the cluster-tree network), while the *top-down* scheme prioritises the downstream traffic. Thus, the choice of the scheduling scheme is an important design issue, because it can directly interfere in the data traffic behaviour of the cluster-tree network.

In order to implement these two scheduling schemes, the `schedulingScheme` simulation parameter was defined (by default, its value is set to 1, which corresponds to a *top-down* scheduling scheme). In case of using a *bottom-up* scheduling scheme, the value of this parameter must be defined as 2 into the parameter configuration file. Table 3.5 shows the parameter used in the cluster scheduling step.

Table 3.5: Parameter used for the cluster scheduling step.

Parameter	Definition
<code>schedulingScheme</code>	Cluster scheduling scheme

Other cluster scheduling schemes can easily be added to the simulation model by

implementing a new version of the *beaconSchedule* function, for example, the TDBS scheme proposed by Koubaa et al. [26].

3.5.1.5 Superframe duration allocation scheme.

The last step of the formation phase is the superframe duration allocation scheme. In a recent work [100], we have shown that IEEE 802.15.4/ZigBee standards do not define the adequate mechanisms to allocate superframe durations for clusters in cluster-tree networks. Therefore, we have conducted a series of experiments showing that the static allocation of superframe durations is not suitable for cluster-tree networks, considering their application requirements and the load imposed by supported message streams.

This way, we have proposed the use of three different allocation schemes, to allocate superframe durations and to setting-up the beacon interval durations for cluster-tree networks, as follows:

1. Standard allocation scheme: the same superframe durations and beacon intervals are allocated for all clusters. In this case, the superframe order is defined by the `beaconOrder` parameter and the beacon interval is defined through the `beaconOrder` simulation parameter. This allocation scheme is static and follows the rules suggested in the IEEE 802.15.4 standard.
2. Proportional-Load allocation scheme: in this scheme, the superframe duration for each cluster is assigned based on the message load imposed by its descendant nodes (the set of descendant nodes includes all nodes along the branches, starting at that cluster-head), improving the throughput of monitoring messages. For this scheme, the beacon interval is set to a value smaller than the smallest message periodicity (respecting the following constraint: $BI = aBaseSuperframeDuration \times 2^{BO}$).
3. Proportional-Node allocation scheme: for this scheme, the superframe duration is assigned based on the number of descendant nodes. This scheme is suitable for applications where the periodicity of supported message streams is unknown. The beacon interval is defined following the same rule of the proportional-Load allocation scheme.

The superframe duration allocation scheme is defined through the `allocationScheme` simulation parameter (by default, its value is set to 1, that corresponds to the standard allocation scheme). For further details about the proportional superframe duration allocation scheme, the reader is referred to [100].

New versions for the superframe duration allocation schemes can be added to the simulation model by implementing new *beaconSchedule* functions. Table 3.6 shows the main parameters used in the superframe allocation scheme step.

3.5.2 Communication phase

The cluster-tree *communication phase* (CP) starts after finishing the formation phase. The CT-SIM communication module can be used to extend available communication protocols or

Table 3.6: Parameters used for the superframe allocation scheme step.

Parameter	Definition
<code>allocationScheme</code>	Superframe allocation scheme
<code>superframeOrder</code>	Superframe order for the clusters
<code>beaconOrder</code>	Beacon order for the clusters

to implement new ones. In order to allow the design and test of new protocols, the basic communication mechanisms of cluster-tree networks were implemented for direct and indirect communication, as defined in the IEEE 802.15.4 standard.

In the direct communication mechanism, nodes directly send their data messages to their parent coordinator. For this, nodes use the slotted CSMA-CA protocol to access the wireless channel. The CSMA-CA mechanism is defined through the following simulation parameters: `macMinBE` (by default, its value is set to 3), `macMaxBE` (by default, set to 5), `macMaxCSMABackoffs` (by default, set to 4), and `macMaxFrameRetries` (by default, set to 3). These parameters can be set up to different values defined at the simulation parameter file.

In the indirect communication mechanism, a coordinator node indicates in the pending address field of its beacon frame that exists pending data. Each child node inspects the beacon frame in order to verify if its address is pending. In positive case, this node requests the data to coordinator during the CAP. In turn, the coordinator acknowledges this request and, thereafter sends the pending data. After receiving the data, the child node confirms the transaction.

Based on this basic communication mechanisms, the proposed simulation model defines two different data traffic types: *monitoring traffic* and *source-to-destination* traffic. *Monitoring traffic* is characterized as the periodic data traffic generated by sensing nodes that is sent towards the sink node of the cluster-tree network (commonly the PAN coordinator). A series of simulation parameters are used to define the monitoring traffic. For example, the data rate is defined by the `packet_rate` simulation parameter, which corresponds to *packets/second*. The data rate can be defined to both a specific node or a set of nodes. We can also prevent nodes from generating monitoring traffic by setting the `dataGen` parameter to *false* (by default, its value is *true*). Besides, the maximum number of data frames can be defined through the `sentPacketMAX` parameter.

More specifically, sensing nodes send monitored traffic to their parent cluster-heads during the cluster active periods, using direct communication mechanisms. Each cluster-head is responsible to provisionally store and forward this traffic towards the sink node, using tree routing. For storing the monitoring traffic of their child nodes, cluster-heads use internal buffers. The size of the internal buffer of a cluster-head can be defined through `macBufferSize` parameter (by default, its value is 32 data frames). Although cluster-heads can perform data aggregation or fusion, CT-SIM still does not provide these mechanisms (they will be implemented in future versions).

Moreover, it has also been defined a *source-to-destination* traffic, which corresponds to a periodic message stream from a source node to a destination node (not necessarily the sink node). In CT-SIM, this traffic is defined in the `omnetpp.ini` configuration file, using specific *Application* parameters. The `source2destination` parameter defines the source and

destination nodes for the data traffic. The `S2DstartupDelay` parameter defines the start time of the message stream. In turn, the `S2DpacketNumber` and `S2DdataRate` parameters define the maximum number of generated packets and its data rate, respectively. The source-to-destination data traffic uses the direct and indirect communication mechanisms.

The simulation model also provides several simulation parameters to configure the physical layer according to a specific radio model, such as: `phyDataRate`, `phyBitsPerSymbol`, `phyDelaySleep2Tx`, and `phyDelayRx2Tx`. Table 3.7 shows the main parameters used during the communication phase.

Table 3.7: Parameters used for the communication phase.

Parameter	Definition
<code>macMinBE</code>	Minimum backoff exponent
<code>macMaxBE</code>	Maximum backoff exponent
<code>macMaxCSMABackoffs</code>	Maximum number of backoffs
<code>macMaxFrameRetries</code>	Maximum number of retries
<code>packet_rate</code>	Data packet rate for the monitoring traffic
<code>dataGen</code>	Allows the generation of monitoring traffic
<code>sentPacketMAX</code>	Maximum number of monitoring frames
<code>macBufferSize</code>	Size of the internal buffer for the cluster-heads
<code>source2destination</code>	Source and destination nodes for the source-to-destination data traffic
<code>S2DstartupDelay</code>	Start time for the source-to-destination traffic
<code>S2DpacketNumber</code>	Maximum number of source-to-destination frames
<code>S2DdataRate</code>	Data packet rate for the source-to-destination traffic
<code>phyDataRate</code>	Radio data rate (kbps)
<code>phyBitsPerSymbol</code>	Number of bits per symbol for the radio
<code>phyDelaySleep2Tx</code>	Delay to switch from the sleep mode to transmission mode
<code>phyDelayRx2Tx</code>	Delay to switch from the received mode to transmission mode

Due to flexibility implementation of simulation parameters, other communication schemes for cluster-tree networks can be easily implemented. By using CT-SIM, it is possible to implement multiple protocols proposed in literature and to develop new communication protocols for wide-scale WSNs. For example, we have used CT-SIM to implement the Alternative-Route Definition (ARound) scheme [98] and the Proportional Superframe Duration Allocation schemes [100].

3.5.3 Statistics and information collection

After the simulation phase, the collection of information is an important phase, where the main statistics about the communication environment and protocols must be handled with special care, allowing network designers to analyse the network behaviour. The statistics and information collection (SIC) phase is responsible to obtain all information about the simulation. For this, CT-SIM uses the available mechanisms provided by Castalia simulator.

CastaliaResults is a special script of Castalia simulator used to interpret simulation results collected through specific source-code functions. For more details about *CastaliaResults* and its collection functions, the reader is referred to the Castalia's manual.⁸ CT-SIM uses the

⁸The Castalia simulator manual: <https://castalia.forge.nicta.com.au/index.php/en/documentation.html>.

CastaliaResults script to obtain informations about the simulation model, in order to allow the interpretation about the main performance metrics, such as: end-to-end delays and packet loss and reception rates. All the statistic functions were directly implemented in the source code. The *CastaliaResults* script is only used to print the results. Table 3.8 summarises several informations that can be obtained from the simulation model.

Table 3.8: Informations about the simulations obtained through *CastaliaResults*.

Parameter	Definition
<i>End-to-end Delay</i>	Delay for the data packets (per each type of traffic)
<i>Packet loss rate</i>	Percentage of lost packets
<i>Packet reception rate</i>	Percentage of successfully received packets
<i>Full buffer</i>	Number of dropped packets due to full buffers of cluster-heads
<i>Generated packets</i>	Number of generated packets (per each type of traffic)
<i>Average in-buffer</i>	Average number of packets in the buffers of cluster-heads
<i>Max in-buffer</i>	Maximum amount of packets achieved during the simulation.
<i>Number of clusters</i>	Number of clusters in the network
<i>Hop number</i>	Hop number of a specific message stream
<i>Consumed energy</i>	Consumed energy during the simulation
<i>Remaining energy</i>	Remaining energy of the nodes

Importantly, *CastaliaResults* script presents the obtained results for each involved node in the simulation. However, this script provides the flexibility of presenting the results as an average of all nodes.

Besides the results obtained through the *CastaliaResults* script, several other information about the simulations can be obtained from text files generated via source code. Among them, we highlight the *Castalia trace* file. Using this file, several important information about all modules can be retrieved after the simulations through the *trace()* function, in which can enumerate: list of cluster-heads and their main information, such as tree depth, list of child leaf nodes and child CH nodes, interference graph, network scheduling, start time for all clusters, parent-child relationships, MAC address mapped to network address, *associatedPAN* for all nodes, buffer behaviour, list of received packets with sequence number, among several other information about the progress of the simulation. These information can easily be obtained and handled through state-of-the-art graphic tools (*gnuplot*, *MATLAB* or *Microsoft Excel*). We have implemented a set of scripts to print results via *MATLAB* tool.

Collected information can be stored into specific files for each network model. Specific functions were implemented to save information about the physical location of all nodes, MAC buffer information and received packets in the application layer. However, other information can be saved into these files by extending the source-code of this function in each network module.

3.6 CT-SIM Simulation experiments

In this section, we present a set of CT-SIM simulation experiments, in order to demonstrate the usage of some of the provided functionalities. For this, the next sections present the different steps of CT-SIM, providing a discussion about the obtained results.

In order to illustrate several of the CT-SIM functionalities, we have set a typical monitoring WSN, with a communication environment set to $200 \times 200 \text{ m}^2$ (`field_x` and `field_y` simulation parameters). The number of nodes (`numNodes` parameter) was set to 201, which corresponds to one PAN coordinator and 200 sensing nodes. The PAN coordinator was placed in the central position, using the `xCoord` and `yCoord` parameters with value defined to 100 m , respectively.

For these simulation experiments, the IEEE 802.15.4-compliant CC2420⁹ radio model was adopted for all nodes. Furthermore, the unit disc model was used as radio propagation model, where the range of the disk was defined to 55 m . Moreover, the used interference model considers that concurrently transmissions generate collisions at the receiver. The simulation results are presented in the following sections.

3.6.1 Node deployment

In order to illustrate the node deployment step, two types of deployment were used: random and grid, with the PAN coordinator located in central position of the environment ($100 \text{ m} \times 100 \text{ m}$). Figure 3.6(a) illustrates this environment using a random deployment for the 200 sensing nodes, while Figure 3.6(b) illustrates the grid deployment generated by the simulation model.

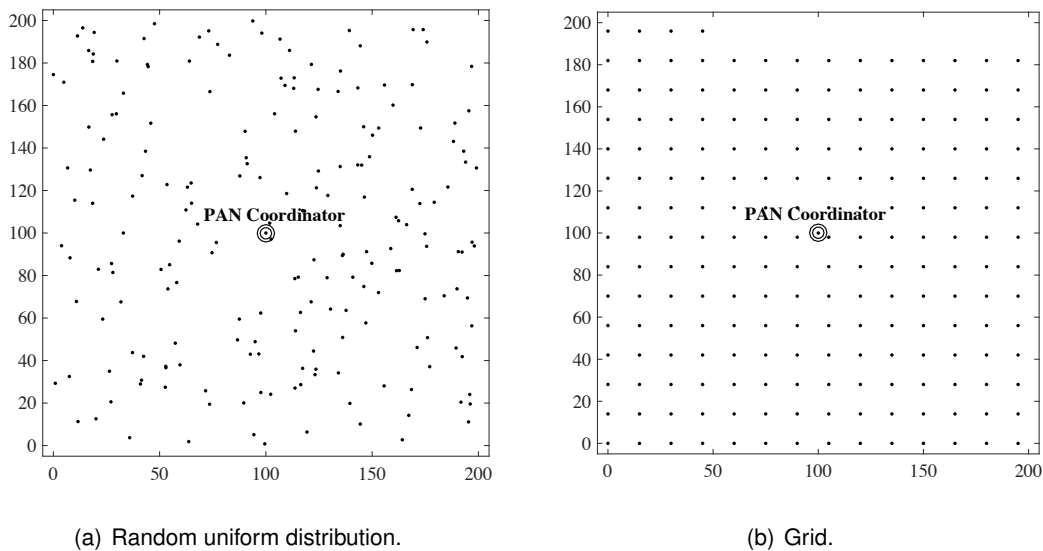


Figure 3.6: Node deployment schemes.

Note that the simulation model is suitable for most WSN applications, where the static, grid and random deployments are commonly used. Most of the available simulators also provides different types of deployments. However, the main advantage of the proposed simulation model is the integration between these different deployments and the other cluster-tree functionalities, such as: network formation, beacon scheduling, superframe allocation schemes and the multiple communication mechanisms.

⁹Texas Instruments / Chipcon CC2420 Datasheet: <http://www.ti.com/product/CC2420/technicaldocuments>.

3.6.2 Cluster formation

Considering the previously defined scenarios, Figure 3.7(a) and Figure 3.7(b) illustrate the formation process of a cluster-tree network. For these experiments, we bounded each cluster to a maximum size of 8 child nodes (`maxChildren` parameter). Also, each coordinator (including the PAN coordinator) can select a maximum of 3 candidate nodes to be cluster-heads (`maxCCH` parameter). Figure 3.7(a) shows a cluster-tree network using a random deployment for the nodes; while Figure 3.7(b) shows a cluster-tree network using a grid deployment.

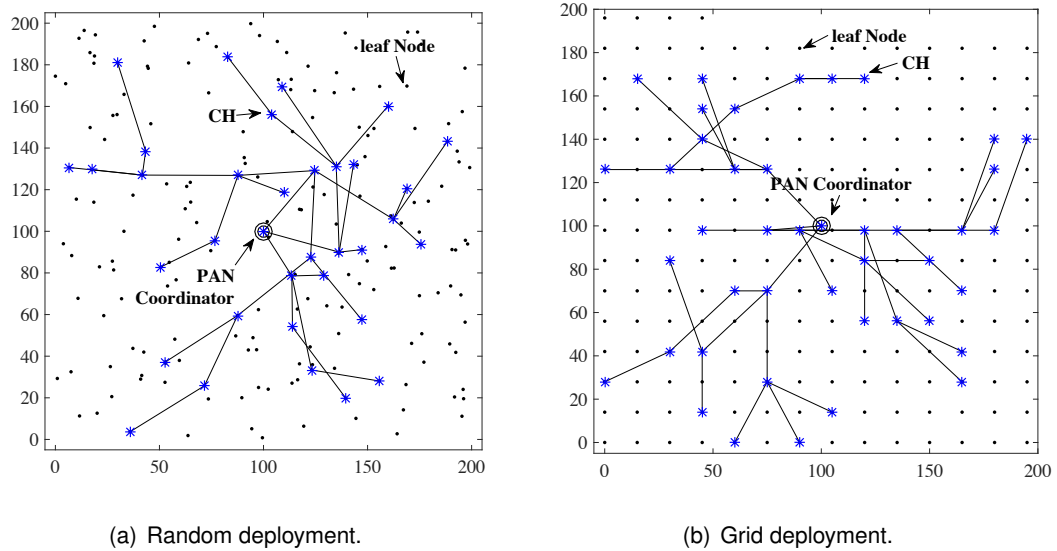


Figure 3.7: Cluster-tree Network.

Table 3.9 presents some information about the cluster-tree network formation for each deployment. Importantly, the simulation environment was fully covered using both deployments and all nodes were associated with a particular cluster-head (no orphan nodes).

Table 3.9: Information about the cluster-tree network formation.

Information	Random	Grid
Number of clusters	35	43
Average number of nodes (per cluster)	6	5
Maximum depth	6	6

3.6.3 Network addressing

For this set of experiments, we have used hierarchical addressing, in which an addressing block was assigned for each cluster-head based on its number of descendant nodes. For the sake of simplification, Table 3.10 shows the addressing information for the cluster-heads of depth 0, 1 and 2, considering the random deployment.

Note that each cluster-head has its own addressing block, in which the first address corresponds to its own network address. It can also be observed that, in this experiments,

Table 3.10: List of addressing blocks of the cluster-heads, considering the random deployment.

CH	Depth	Network address	Addressing Block
0	0	0	[0, 200]
4	1	1	[1, 106]
9	1	107	[107, 145]
27	1	146	[146, 195]
8	2	2	[2, 21]
28	2	22	[22, 53]
35	2	54	[54, 101]
68	2	108	[108, 118]
83	2	119	[119, 126]
103	2	127	[127, 140]
22	2	147	[147, 154]
42	2	155	[155, 185]
86	2	186	[186, 190]

cluster-heads do not have available addresses to allow new node associations. The hierarchical addressing allows deterministic routing, because each cluster-head decides if a received packet must be forwarded to the parent cluster-head or to its child nodes based on its addressing block.

3.6.4 Cluster scheduling

In order to illustrate the cluster scheduling, Table 3.11 shows the final scheduling for the top-down and bottom-up cluster scheduling, considering the random deployment. For the sake of convenience, we defined that the cluster ID corresponds to the cluster-head ID.

Table 3.11: Cluster scheduling for the random cluster-tree network experiment.

CH	Depth	Top-down	Bottom-up
0	0	1 st	35 th
4	1	2 nd	34 th
9	1	3 rd	33 th
27	1	4 th	32 th
8	2	5 th	31 th
28	2	6 th	30 th
35	2	7 th	29 th
68	2	8 th	28 th
83	2	9 th	27 th
103	2	10 th	26 th
22	2	11 th	25 th
42	2	12 th	24 th
86	2	13 th	23 th
⋮	⋮	⋮	⋮
56	4	32 th	4 th
92	4	33 th	3 rd
116	5	34 th	2 nd
101	5	35 th	1 st

Note that, in the top-down scheduling, the first selected cluster corresponds to the PAN coordinator's cluster, followed by clusters of depth 1, and, finally, clusters of depth 2. In the bottom-up scheduling, the deepest clusters are firstly selected. Then, clusters of upper depth are lastly selected. It can be observed that the PAN coordinator's cluster is the last cluster in the scheduling list.

3.6.5 Superframe duration allocation

An important CT-SIM functionality is to provide different superframe duration allocation schemes. To the best of our knowledge, no other simulator or simulation model provides this functionality for general WSN simulation. We have shown that adequate superframe duration allocation is a crucial issue for several WSN applications [100]. The IEEE 802.15.4 standard does not provide any mechanism to adequately allocate values for superframe order and beacon order for cluster-tree networks. In order to illustrate this functionality, and for the sake of simplicity we presented some obtained results through the experiments just for the random environment.

Table 3.12 shows the obtained results by applying a static superframe allocation scheme. In this allocation scheme, all clusters have the same SO parameter defined by `beaconOrder` simulation parameter.

Table 3.12: Static superframe duration allocation scheme for the random cluster-tree network experiment.

CH	Depth	SO	SD (symbols)
0	0	3	7680
4	1	3	7680
9	1	3	7680
27	1	3	7680
8	2	3	7680
28	2	3	7680
35	2	3	7680
68	2	3	7680
83	2	3	7680
103	2	3	7680
22	2	3	7680
42	2	3	7680
86	2	3	7680

In turn, Table 3.13 shows the obtained results by applying a proportional superframe allocation scheme. In this allocation scheme, the superframe durations are defined based on the load imposed by descendant nodes (nodes belonging to the branch of the cluster-head). The main idea is to allocate adequate bandwidth for each cluster-head in order to improve the network throughput.

Furthermore, a proportional superframe allocation scheme can also allocate adequate buffer size for all cluster-heads. The main idea is to dimension a proper buffer size for each cluster-head based on the message load, allowing to save resources of the nodes. Table 3.14 shows the size of internal buffers (number of messages) using both the static and proportional superframe duration allocation schemes.

Table 3.13: Proportional superframe duration allocation scheme for the random cluster-tree network experiment.

CH	Depth	SO	SD (symbols)
0	0	7	122880
4	1	6	61440
9	1	4	15360
27	1	5	30720
8	2	3	7680
28	2	4	15360
35	2	5	30720
68	2	3	7680
83	2	2	3840
103	2	3	7680
22	2	2	3840
42	2	4	15360
86	2	1	1920

Table 3.14: Size of internal buffers for the cluster-heads using different superframe duration allocation schemes.

CH	Depth	Static scheme	Proportional scheme
0	0	200	201
4	1	200	106
9	1	200	39
27	1	200	50
8	2	200	20
28	2	200	32
35	2	200	48
68	2	200	11
83	2	200	8
103	2	200	14
22	2	200	8
42	2	200	31
86	2	200	5

Note that for the static allocation scheme, we have defined the size of buffers equal to the number of sensing nodes in the cluster-tree network, without considering any features of the cluster-heads, such as their depths and number of child nodes. In turn, the proportional allocation scheme presents different results by considering the requirements of the network formation process.

The proposed CT-SIM simulation model provides information about the behaviour of the internal buffers for all cluster-heads, such as the average number of messages within the buffers during the active periods of the clusters (via *CastaliaResults* script). To illustrate this functionality, Figure 3.8 shows the behaviour of the buffers for different depths of the cluster-tree, using the top-down cluster scheduling; while Figure 3.9 shows the behaviour of the buffers, using the bottom-up scheduling.

Note that the buffer behaviour is an important metric to analyse the network performance. It can be observed in Figures 3.8 and 3.9 that a proportional superframe duration allocation scheme has better performance behaviour when compared with a static scheme. Besides to

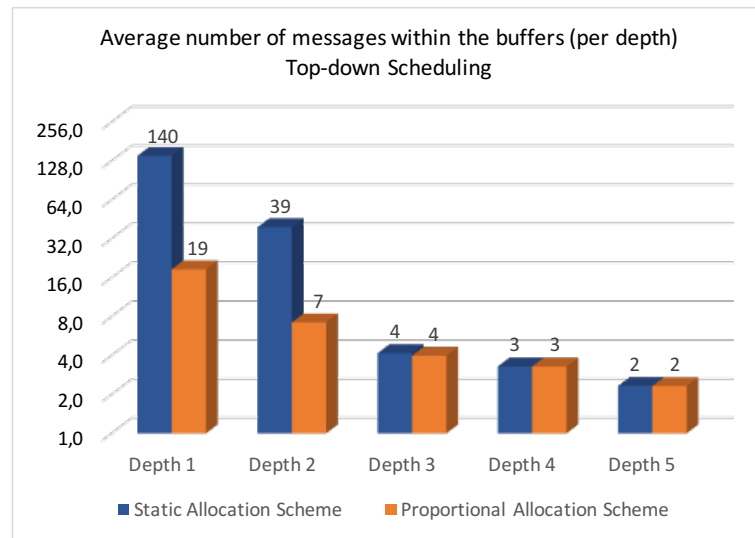


Figure 3.8: Average number of messages within the buffers of the cluster-heads (per depth), using top-down cluster scheduling.

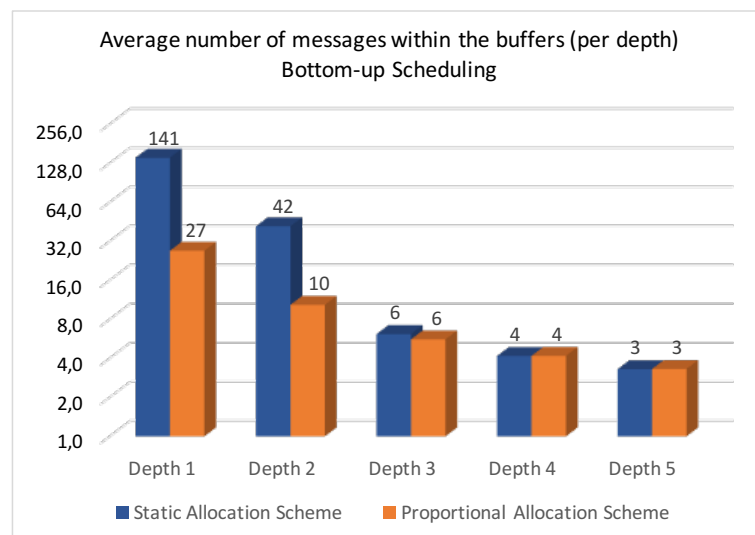


Figure 3.9: Average number of messages within the buffers of the cluster-heads (per depth), using bottom-up cluster scheduling.

save resources, an adequate superframe allocation can avoid the accumulation of messages within the buffers, which generate discard of messages due to buffer overflows and high delays. For example, Figure 3.10 shows the number of discarded messages due to buffer overflows.

As can be seen in Figure 3.10, the proportional superframe duration allocation scheme may avoid the discard of messages due to buffer overflows; while the static scheme has a high number of discarded messages within the buffers, increasing the message loss rate and the end-to-end delays. This performance can be unsuitable for a large number of WSN applications. Thus, it is very important that a simulation model provides mechanisms to evaluate the buffer behaviour of cluster-heads in cluster-tree WSNs.

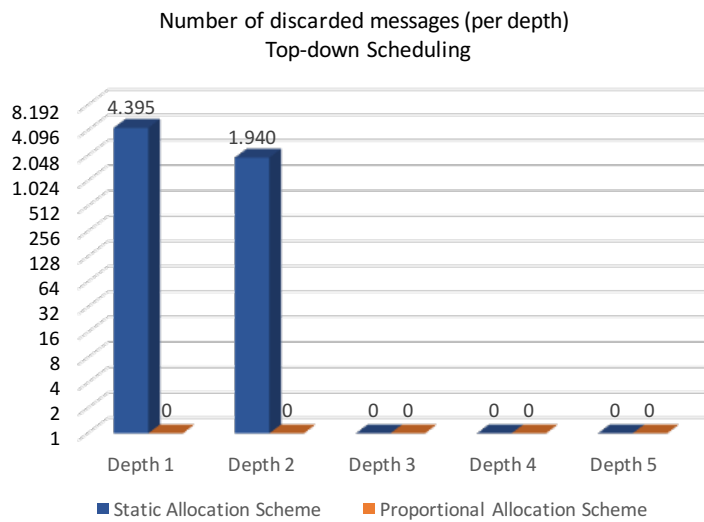


Figure 3.10: Number of discarded messages within the buffers of the cluster-heads (per depth), considering the top-down cluster scheduling.

3.6.6 Communication phase

In order to demonstrate the usability of CT-SIM in what concerns the support of different communication traffics, we have performed another set of simulations. First, we performed a simulation to demonstrate the behaviour of the monitoring traffic under different number of nodes and data rates. Table 3.15 shows the different scenarios defined for this experiment.

Table 3.15: Set of scenarios to evaluate the communication phase.

Scenario	Number of nodes	Environment size
Scenario 1	50 + PAN	100m x 100m
Scenario 2	100 + PAN	141m x 141m
Scenario 3	150 + PAN	173m x 173m
Scenario 4	200 + PAN	200m x 200m

For all scenarios, the PAN coordinator was placed at the central location of the environment, and the other sensing nodes were randomly deployed. Note also that, as the number of nodes increase, the environment size proportionally increases in order to keep the same node density. The basic idea is to avoid biased assessments due to the collision effect caused by different node density.

Regarding to data rates, we have defined two types of the monitoring traffic: 0.1 pkt/s (1 packet every 10 s) and 0.05 pkt/s (1 packet every 20 s). For each scenario, data rates were equally distributed among the sensing nodes. The PAN coordinator is the unique node that does not generate data traffic. Each node generates 50 messages. In order to provide diversity of the results, each scenario runs 5 times using different sets of random variables.

Figure 3.11 shows the average end-to-end delay for the monitoring traffic, considering the static and proportional superframe duration allocation schemes.

Note that, as the number of nodes increase, the average end-to-end delay also increases due to a higher load of messages. However, we can observe that a proportional superframe

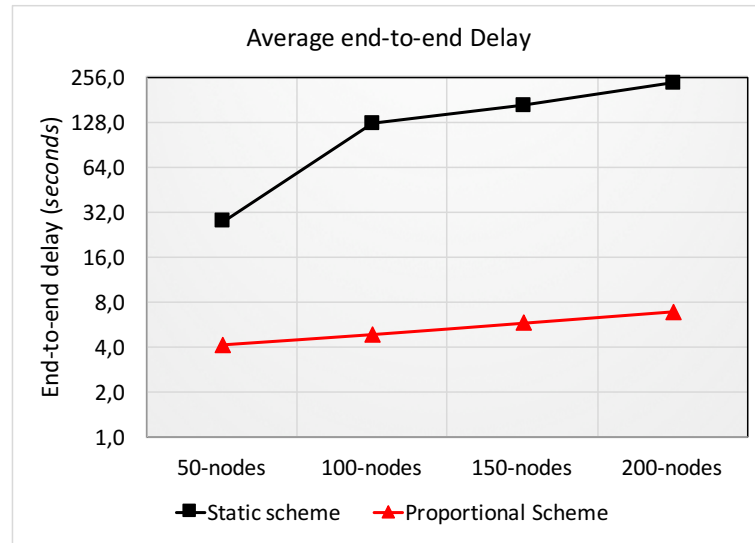


Figure 3.11: Average end-to-end delay for the monitoring traffic under different number of sensing nodes.

duration allocation scheme has a better performance, because network parameters were adequately defined according to the number of nodes and the message load imposed by them. Instead, a static allocation scheme does not consider the different conditions for the cluster-heads to allocate superframe durations and buffer sizes, and, consequently, the message delays highly increase.

Additionally, aiming to demonstrate the usability of the proposed simulation model regarding to the source-to-destination traffic, an experiment was carried out using both the two available traffic types together. For this, we have defined an environment size of $200\text{ m} \times 200\text{ m}$. Again, the PAN coordinator was deployed in central position. In turn, nodes 1 and 2 were defined as the source and destination nodes for the source-to-destination traffic, respectively. These nodes were statically deployed in opposite positions with the PAN coordinator placed in the middle. Regarding the source-to-destination traffic, node 1 generates 1000 packets towards node 2 with data rate set to 0.2 pkt/s (1 packet every 5 s). Importantly, nodes 1 and 2 do not generate monitoring traffic. Moreover, we randomly deployed 400 sensing nodes, which generate monitoring traffic. In order to obtain diverse scenarios, we have defined different monitoring traffic data rates: 0.05 pkt/s (1 pkt/20s), 0.025 pkt/s (1 pkt/40s), 0.0166 pkt/s (1 pkt/60s) and 0.0125 pkt/s (1 pkt/80s).

Figure 3.12 illustrates this simulation environment and shows the tree path that supports the cluster-tree communication from node 1 to node 2 (source-to-destination traffic).

Figure 3.13 shows the average end-to-end delay for the monitoring and source-to-destination traffics. Note that, as the monitoring traffic rate increases, the average end-to-end delay also increases. In fact, increasing the data rate for the monitoring traffic, the number of packets waiting to be transferred in the MAC buffers also increases, inducing higher delays for the packets. Also, there is a large number of nodes trying to contend for the wireless channel. Note also that, the source-to-destination traffic keeps a constant end-to-end delay (in general) while the data rate increases until the 1 pkt every 40s for the monitoring traffic. However, using the data rate of 1 pkt every 20s for the monitoring traffic, the average end-to-end delays increase for both the monitoring and source-to-destination

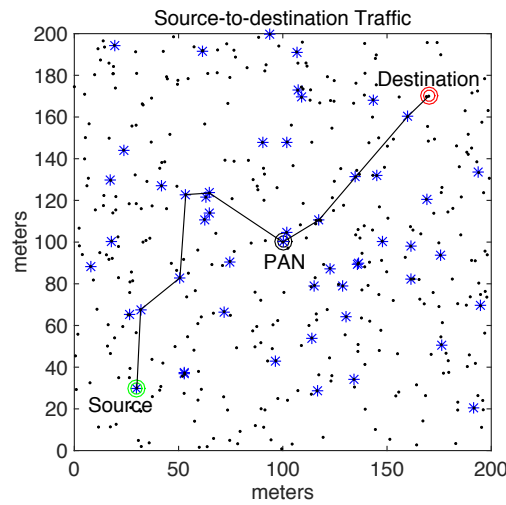


Figure 3.12: Tree routing for the source-to-destination traffic.

traffics, which shows that the cluster-tree can be saturated.

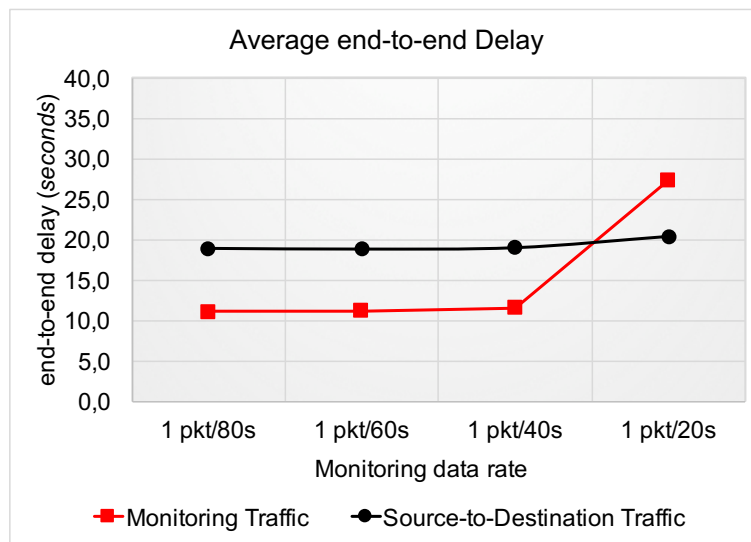


Figure 3.13: Average Delay for the monitoring and source-to-destination traffics.

3.7 Conclusion and future considerations

In this paper, we presented CT-SIM, a simulation model proposed for IEEE 80215.4/ZigBee-based cluster-tree WSNs, that was built-upon the Castalia Simulator. This simulation model allows to build random wide-scale cluster-tree networks, considering its most important mechanisms and issues, such as: network formation, addressing, cluster scheduling, superframe duration allocation and, direct and indirect data communication mechanisms. We demonstrate through a series of simulation experiments the main features and capabilities of the proposed CT-SIM simulation model.

CT-SIM was designed in a modular way, which allows the configuration of a multiple communication environments and network parameters. With this, researchers and developers can design and evaluate new protocols and implementations for wide-scale cluster-tree networks based on the IEEE 802.15.4/ZigBee standards.

As future considerations, we intend to add new functionalities and algorithms for the proposed simulation model and to implement new communication protocols and approaches for wide-scale cluster-tree networks. Also, we intend to make an evaluation version of this simulation model, that will be available for the research community.

This page was intentionally left in blank

Superframe Duration Allocation Schemes to Improve the Throughput of Cluster-tree Wireless Sensor Networks

This chapter presents a set of Proportional Superframe Duration Allocation (SDA) Schemes, whose target is to improve the network throughput, avoiding congestion, higher end-to-end communication delays and discarded messages due to buffer overflows. This chapter is a reproduction of the following manuscript:

LEÃO, Erico; MONTEZ, Carlos; MORAES, Ricardo; PORTUGAL, Paulo; VASQUES, Francisco. Superframe Duration Allocation Schemes to Improve the Throughput of Cluster-tree Networks based on IEEE 802.15.4 and ZigBee Standards. **SUBMITTED TO:** *Sensors*, MDPI, 2016.

4.1 Abstract

The use of Wireless Sensor Network (WSN) technologies is an attractive option to support wide-scale monitoring applications, such as those that can be found in precision agriculture, environmental monitoring and industrial automation. The IEEE 802.15.4/ZigBee cluster-tree topology is a suitable topology to build wide-scale WSNs. Despite its advantages, such as timing synchronisation, duty-cycle operation, and beacon-enabled mode, cluster-tree networks forward all communication through the tree paths via cluster-head nodes, toward the PAN coordinator. Thus, the careful adjustment of the length of the transmission opportunities allocated to each of the cluster-heads (superframe durations) is an important research issue. This paper proposes a set of proportional Superframe Duration Allocation (SDA) schemes, based on well-defined protocol and timing models for IEEE 802.15.4 cluster-tree networks, that consider the message load imposed by child nodes (*Load-SDA* scheme) and the number of descendant nodes (*Nodes-SDA* scheme) of each cluster-head. The underlying idea is to adequately allocate superframe durations and buffer sizes to each of the cluster-heads, in order to improve the network throughput and to avoid typical problems, such as: congestion, high end-to-end delays and discarded messages due to buffer overflows. Simulation assessments show that the proposed allocation schemes may improve the operation of cluster-tree networks and avoid their inherent problems.

4.2 Introduction

With the increasing technological advances of Micro-Electro-Mechanical devices [10], including its processing and storing capabilities, Wireless Sensor Networks (WSN) have become an attractive technology to deploy wide-scale industrial and home applications, such as: environmental monitoring, precision agriculture, smart buildings and cities, industrial automation and military applications [5, 7].

WSNs are special wireless *ad hoc* networks composed of a large number of low-power, low-cost and low-rate devices, which are capable of sensing, processing and sending information related to environment variables [86]. This type of network may also be able to actuate over the monitored environment through the use of special devices called actuators. The increasing demand for WSN-based applications is driving the need for new design approaches, able to deal with WSN specific requirements, such as energy-efficient operation, wide-scale deployments and time-sensitive approaches.

The IEEE 802.15.4/ZigBee standards [18, 21] are the most widely used protocols to deploy WSNs. On one hand, the IEEE 802.15.4 standard defines the Physical layer and Medium Access Control (MAC) sublayer for low-rate, low-cost, and low-power Wireless Personal Area Network (LR-WPAN) applications. On the other hand, ZigBee specifies the upper layers (Network and Application) over the IEEE 802.15.4 protocol stack. Basically, these standards define two types of devices: Full-Function Devices (FFDs) and Reduced-Function Devices (RFDs). While RFDs perform only a reduced set of functions such as channel scanning, network association requests and sensing activities, FFDs can also perform more complex functions such as PAN

Depending on the application requirements, an IEEE 802.15.4 network can operate in two topology types: star and peer-to-peer. The star topology is the simplest network organisation, where all sensing nodes are directly connected to the PAN coordinator, which is a unique node and is responsible for all management and communication activities (centralized communication paradigm). Although being easy to build and manage, the weakness of this topology is that the network coverage is limited to the sensing range of its nodes, which prevents wide-scale deployments.

In peer-to-peer topologies, any device may directly communicate with any other device, as long as they are in the communication range of one another. This topology type allows more complex network formations to be implemented, such as cluster-tree and mesh topologies. Mesh topologies provide higher network flexibility and lower complexity, high routing redundancy and good scalability [25, 26]. However, it does not provide explicit mechanisms that allow nodes to enter into low power mode [27, 28], decreasing the network lifetime. These characteristics are not desirable for typical WSN-based monitoring applications, which imposes strict requirements regarding to power consumption of the sensor nodes.

In order to overcome this weakness, the IEEE 802.15.4/ZigBee set of standards also provides the cluster-tree topology – a special peer-to-peer topology. Cluster-tree is one of the most suitable topologies to deploy wide-scale WSNs [30]. In the cluster-tree topology, nodes are grouped into clusters and are coordinated by a unique FFD node called cluster-head (the PAN coordinator is a specific case of a cluster-head). Cluster-heads are responsible for association, synchronisation and communication of their child nodes. In order to provide

scalability, clusters are interconnected through their coordinators to build a hierarchical network structure.

Nevertheless, the efficient operation of cluster-tree topologies requires the consideration of further design issues, such as: network formation [44, 48], beacon scheduling [26, 52], and MAC protocol configuration related issues [55, 56, 64, 65], including those related to medium access protocol and those that define the communication structures. Thus, it is necessary to provide adequate guidelines for setting-up the MAC configuration parameters in order to build efficient wide-scale cluster-tree WSNs.

4.2.1 Objectives and Contributions

In this paper, we define a holistic approach to deal with the problem of how to allocate superframe durations to the multiple cluster-heads in a wide-scale WSN. This paper extends work previously presented in [100]. We define a set of boundary equations that model both the protocol and the timing behaviour of IEEE 802.15.4 cluster-tree networks in worst-case conditions. Based on these equations, we propose a set of superframe duration allocation schemes, that are able to improve the network throughput in wide-scale WSN applications. The proposed allocation schemes distribute the available network bandwidth according to the traffic associated with each network cluster. As a consequence, the use of this type of allocation schemes enables the reduction of some of the typical problems of cluster-tree networks such as: congestion, high delays and dropped messages due to buffer overflows. Basically, the proposed superframe duration allocation schemes consider: 1) the message load imposed by sensor nodes; 2) the number of descendant nodes of each cluster; and 3) the number of child nodes belonging to the cluster itself. This type of bandwidth allocation scheme is based on earlier work developed to set the bandwidth allocation of Fieldbus networks [101] and FDDI networks [102] and, more recently, of FlexRay networks [103, 104].

We focus our study on ZigBee-based cluster-tree topologies, due to some specific features, such as: suitability to deploy wide-scale networks and energy-efficiency guarantees, which are common features for typical WSN monitoring applications.

To the best of our knowledge, this is the first study that provides guidelines for setting the superframe structure in cluster-tree networks, considering the timing constraints imposed by the traffic generated by the sensor nodes and the network topology. The main contributions of this paper can be summarized as follows:

- A set of worst-case boundary equations for IEEE 802.15.4 cluster-tree networks, defining a set of timing constraints that must be fulfilled during the network operation, in order to guarantee the adequate timing behaviour of the supported applications.
- A set of guidelines to define the superframe duration and the buffer size for each cluster. These guidelines are based on both the traffic requirements and the network topology, such as the number of child nodes and the depth of the cluster-heads in the cluster-tree network. The use of these guidelines for setting-up the superframe durations significantly increase the network throughput, avoiding or reducing the congestion of the cluster-tree network.
- A simulation assessment that highlights the advantages of using the proposed superframe duration allocation schemes.

4.2.2 Organisation of this Paper

The remainder of this paper is organized as follows: Section 4.3 provides the required background. Subsection 4.3.1 presents an overview of the IEEE 802.15.4/ZigBee cluster-tree networks and Subsection 4.3.2 presents some of the most relevant related works for the development of this work. Section 4.4 defines the considered message traffic and network models. Section 4.5 presents a set of boundary equations that will constrain the allocation of the superframe durations for each cluster-head. In Section 4.6, we model the considered transmission duration time considered in this work. Section 4.7 presents a timing constraint for the monitoring traffic based on the protocol constraints of IEEE 802.15.4 cluster-tree networks. Section 4.8 introduces the proposed superframe duration allocation schemes. Subsection 4.8.1 presents an allocation scheme based on the load imposed by the descendant nodes of each cluster-head; Subsection 4.8.2 presents an allocation scheme based on the number of descendant nodes; Subsection 4.8.3 provides an example of the use of the proposed allocation schemes. Finally, Section 4.9 presents a simulation assessment of the proposed allocation schemes and discussion of the results, and some conclusions and considerations about future works are presented in Section 4.10.

4.3 Background

4.3.1 Cluster-tree Topologies

Cluster-tree topologies are complex peer-to-peer constructions, where sensor nodes are grouped into clusters. Each cluster is coordinated by a specific FFD node called coordinator or cluster-head (CH). All communication within the clusters is centralized under the control of the CH. The CH is responsible to build its own cluster, managing nodes' association and providing synchronisation mechanisms and intra-cluster communication.

The first cluster of a cluster-tree network is initially built by a special node, called PAN coordinator. The PAN coordinator is responsible for all the network management activities and normally is the sink node of the network. The CHs (including the PAN coordinator) are interconnected by parent-child relationships, forming a tree hierarchical network structure (multicluster).

Cluster-tree networks operate in the IEEE 802.15.4 beacon-enabled mode, where a structure called *Superframe* organizes all communication rounds. A superframe is bounded by beacon frames, which are periodically transmitted by the cluster-heads. Beacon frames are used to synchronise and to identify the clusters and also to describe the superframe structure. The superframe structure is described by two parameters: the *macBeaconOrder* (BO) and *macSuperframeOrder* (SO), where $0 \leq SO \leq BO \leq 14$. These parameters define the *Beacon Interval* (BI) and the *Superframe Duration* (SD), respectively. Figure 4.1 illustrates the superframe structure.

BI defines the interval at which coordinators must periodically transmit their beacon frames. In turn, SD defines the communication period of the clusters. Each superframe can be composed of two parts: the active and inactive periods. The inactive period exists only when the SO parameter is smaller than the BO parameter. During the active period, nodes

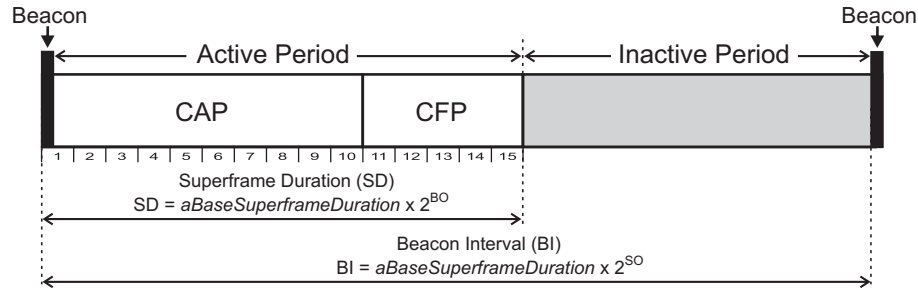


Figure 4.1: IEEE 802.15.4 Superframe structure.

can communicate with their cluster-heads. On the other hand, during the inactive period, the coordinator and member nodes may enter in low power (sleep) mode in order to save energy. The active part comprises two periods: *Contention Access Period* (CAP), during which member nodes can communicate using slotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) mechanism to access the channel; and *Contention-Free Period* (CFP), during which the coordinator can allocate up to seven *Guaranteed Time Slots* (GTS) for specific devices to transmit data without contending for the channel access. The *aBaseSuperframeDuration* parameter defines the minimum duration of a superframe (SD_{min}) when SO is 0 (by default, this parameter corresponds to 960 symbols, corresponding to a duration of 15.36 ms, considering a bit rate of 250 kbps, frequency band of 2.4 GHz, and one symbol as 4 bits).

4.3.2 Related Works

In recent years, multiple works have been presented addressing some of the main challenges about cluster-tree WSNs, including network formation schemes, communication mechanisms, MAC protocol configuration, energy-efficiency, scalability issues, admission and congestion control, and beacon scheduling. Each of these issues has its own special considerations. Within the context of this paper, we are particularly interested in works that address the throughput of the network traffic for IEEE 802.15.4 cluster-tree WSNs, using CAP communication mechanisms. For this topic, modelling the main constraints of the IEEE 802.15.4 MAC protocol using formal methods and the definition of adequate communication period allocation schemes are important mechanisms to avoid known problems such as packet drops due to buffer overflows, congestion and high end-to-end delivery delays, which could lead to undesired operation of the network [99]. Several works in the literature show that the configuration of IEEE 802.15.4 MAC parameters has a direct impact upon the performance of the WSN, which can generate undesirable behaviours for applications subject to strict requirements, such as: energy efficiency, wide-scale deployments and time-sensitive message transfers [56, 65].

In this context, we point out a set of works [105–108] that provide analytical models for the contention access period of IEEE 802.15.4 MAC protocol. Cao et al. [105] presents an accurate analytical model to evaluate the behaviour of IEEE 802.15.4 MAC protocol with periodic traffic, which is a common scenario for WSN-based monitoring applications. The authors consider the probabilities of CCA (Clear Channel Assessment) failures and transmission collisions, considering the standard characteristics of retransmissions and the

double CCA of the CSMA-CA protocol. Although the authors point out that this model can be used to define adequate active period duration, no scheme or guidelines are provided.

Several works encompass analytical models of the contention period for the IEEE 802.15.4 MAC protocol based on Markov chain [106–108]. Basically, each one of these analytical models consider a specific set of parameters and characteristics of the CSMA-CA protocol. Guennoun et al. [106] provided a new IEEE 802.15.4-based MAC protocol named *Variable CCA MAC protocol*. The main idea behind this protocol is to change the number of CCAs that a node must perform before transmitting a data packet (by default, it is defined to 2). The authors modelled this new MAC protocol using Markov chains and demonstrate its accuracy and capability of predicting its behaviour through simulation. However, this new protocol has a negative performance regarding to channel utilisation communication, delays, reliability and energy consumption. In fact, increasing the number of CCAs leads to the increasing of the contention window, which can generate higher delays and energy consumption. Furthermore, performing a higher number of CCAs does not avoid collisions, since nodes are not spread along the time to access the wireless channel. Recas et al. [107] proposed an analytical model based on Markov chains, considering several node classes, by setting different values for the CSMA-CA protocol such as contention window, maximum number of backoffs and the initial and maximum backoff exponent. However, this work does not consider inactive periods. Instead, Park et al. [108] provided two Markov chains to model the behaviour of IEEE 802.15.4 MAC protocol, considering the contention period (CAP) and the contention free period (CFP). This analytical model considers the main parameters of CSMA-CA protocol and the network performance is evaluated through both theoretical analysis and simulation.

A major drawback of these approaches is that they only consider star networks and do not address the characteristics of cluster-tree networks. Thus, these approaches are limited to specific environments and can not be applied to wide-scale applications. In addition, other weakness of these works is that they provide analytical models for the IEEE 802.15.4 MAC protocol without considering any scheme to adequately allocate the active period durations to improve the throughput for data traffic.

Other works addressed analytical models encompassing cluster-tree networks [42, 109, 110]. Martalò et al. [109] proposed an analytical framework to model the behaviour of the IEEE 802.15.4 MAC protocol based on Markov chains. An important requirement assumed by the authors is the finite buffer queues for the nodes. The provided model is quite simple, where the node traffic is always generated by nodes and forwarded toward sink node (PAN coordinator), and it does not describe the complex features of this network type. Jurcík et al. [42] used network calculus theory to model cluster-tree WSNs according to several network parameters, such as: depth, maximum number of child nodes and maximum number of child routers. In this work, the authors provide a worst-case evaluation of the behaviour for upstream and downstream data flows, considering important constraints such as: buffer and bandwidth requirements, flow directions and end-to-end delays. However, this work considers only the contention-free period (GTS), and the limitation of maximum of seven GTS allocation imposed by IEEE 802.15.4 standard restricts the number of data flows in network. Moreover, the data traffic generated by typical monitoring scenarios was not considered in this analysis. Kohvakka et al. [110] provided several mathematical models for the performance analysis of IEEE 802.15.4 CSMA-CA mechanism and MAC operations and verified the proposed models through simulations.

However, only models were provided and no further schemes were proposed.

Moreover, we highlight other set of works proposed by [28, 111], which provide performance assessments of IEEE 802.15.4 MAC protocol by setting different values for its MAC parameters. Di Francesco et al. [28] analysed the impact of the main MAC parameters upon the communication behaviour and proposed an adaptive cross-layer framework to minimize the energy consumption for single and multihop WSNs. The authors have shown that changing the *macMinBE*, *macMaxCSMABackoffs* and *macMaxFrameRetries* parameters may increase the probability of winning the contention for the wireless channel. However, increasing these values above a certain threshold does not interfere significantly. Also, they have shown that the *macMaxCSMABackoff* parameter has higher impact over the energy consumption than *macMinBE* parameter. Chen et al. [111] provided a performance evaluation of IEEE 802.15.4 star networks through simulation. Differently of [28], the authors focused on selection of the SO and BO parameters and their impact upon different industrial scenarios. However, only star topologies were evaluated and no allocation scheme was proposed.

Some works have used other techniques to increase the bandwidth for message streams, such as: beacon scheduling [52–54] and superframe duration adjustment scheme [56, 57, 112]. Hanzalek and Jurcík [52] presented a Time-Division Cluster Scheduling (TDCS) mechanism to meet end-to-end deadlines of time-bounded message streams. This mechanism employed a pure time-division scheduling approach, avoiding the inter-cluster collision problem. The authors formulated the TCDS approach as a cyclic extension of the Resource Constrained Project Scheduling with Temporal Constraints (RCPS/TC), which defines a feasible schedule considering the temporal and resource constraints for a set of tasks. After modelling this problem, they use an integer linear programming algorithm to solve the scheduling problem. Severino et al. [53] proposed a dynamic cluster scheduling scheme to provide QoS for different traffic flows in cluster-tree networks. In this work, the authors defined a run-time approach to re-order the involved clusters in specific message streams, considering their priorities, in order to minimize the traffic latency. Also, this approach provides a mechanism to increase the size of the superframe duration (bandwidth) of the involved clusters in the communication of message streams, using the inactive period or the active periods of non-involved clusters. Yeh and Pan [54] formulated the Low-latency Two-way Beacon Scheduling (LTBS) problem for cluster-tree networks. In this approach, the authors modify the IEEE 802.15.4 superframe structure to allow the broadcast of two beacons: one active part for the upstream traffic and another for downstream traffic. The authors also defined a set of algorithms to assign nodes in upstream and downstream slots in order to reduce the network latency, avoiding interferences among them.

Furthermore, Lee et al. [56] provided a Superframe Duration Adjustment Scheme (SUDAS) based on Markov chains, which analyses both the contention and contention-free periods, allocating GTS slots for devices based on the packet sizes. The main idea of SUDAS is to adequately allocate GTS and define the start time for a set of requested devices, improving the bandwidth of the contention-free period. With this, the contention period can be used by other nodes to communicate their data packets. Rasouli et al. [57] proposed an algorithm for the adjustment of the length of superframe duration and the CSMA Backoff Exponent (BE) parameter according to the network traffic, in order to decrease the energy consumption and to improve the throughput of the network. However, these schemes

just consider star topologies. Casilari et al. [112] provided algorithms to define the superframe durations for all clusters in a cluster-tree network, following a time-division approach. The main idea is to maximise the use of the beacon interval and to avoid any inactive period. For this, the authors proposed different allocating schemes, such as: the same SO for all clusters, highest SO for the PAN coordinator and a scheme that allocates a SO for the coordinator based on the traffic generated in its cluster. However, the authors just present simple schemes and do not consider several protocol constraints of cluster-tree, for instance, buffer constraints and timing constraints of messages.

Within this context, we can easily observe the lack of mechanisms to properly allocate active communication periods within the CAP, in order to improve the throughput for monitoring traffic in cluster-tree networks, avoiding common problems that can lead to undesirable network states (e.g. buffer overflows). This paper aims to provide a set of guidelines that enable network designers to efficiently predict appropriate network operational parameters and message flow configurations for wide-scale cluster-tree networks.

4.4 System Model

In this work, we assume a set of N_{nodes} sensor nodes organized according to a cluster-tree topology and randomly deployed along a wide-scale environment. Moreover, we consider that the network formation procedure ensures that all monitoring environment is covered (no orphan nodes). The PAN coordinator (root of the tree and sink node) is responsible to trigger the network formation and acts as cluster-head for the first cluster, according to the IEEE 802.15.4 [68] and ZigBee [21] standards.

Moreover, the cluster-tree network is composed of a set of N_{CH} coordinator nodes (including the PAN coordinator), acting as cluster-heads (CHs) of their clusters and periodically sending beacon frames to synchronise their child nodes. We assume that this set of clusters is static and that there are no mobile nodes, which imply that the network topology does not change along the network lifetime. Figure 4.2 illustrates an IEEE 802.15.4 cluster-tree wireless sensor network deployed in a specific wide-scale environment.

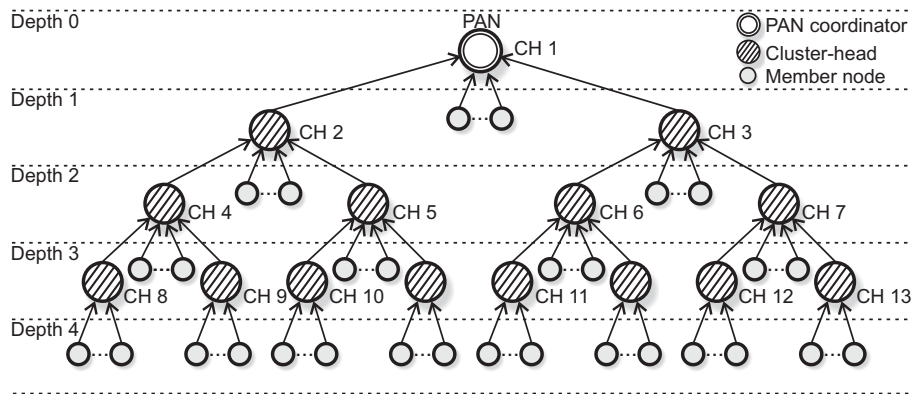


Figure 4.2: IEEE 802.15.4/ZigBee cluster-tree network.

Within this context, we consider that cluster-heads share the same Beacon Interval (BI),

but a specific superframe duration (SD) is defined for each one of them. Therefore, the set of cluster-heads is characterized by:

$$CH_j = (SD_j, BI), \quad \text{for } 1 \leq j \leq N_{CH}, \quad (4.1)$$

where SD_j is the Superframe Duration of the cluster-head CH_j and BI is the Beacon Interval for all clusters.

To avoid inter-cluster collisions caused by overlapping clusters, we assume that all cluster's active periods have been previously scheduled according to a time division beacon scheduling approach. For this, two different beacon scheduling schemes may be adopted:

- Top-down beacon scheduling: superframe durations are ordered in a top-down direction. Firstly, the PAN coordinator is scheduled, then clusters of the depth 1, and later clusters of the following depths are sequentially aligned. This scheduling scheme prioritises downstream traffic.
- Bottom-up beacon scheduling: superframe durations are ordered in a bottom-up direction. Firstly, the deepest clusters (max depth of the tree) are scheduled, then clusters of next lower depth, following depth-by-depth until reaching the PAN coordinator. This scheduling scheme prioritises upstream traffic.

Figure 4.3 illustrates a top-down beacon scheduling, while Figure 4.4 illustrates a bottom-up scheduling for the network presented in Figure 4.2. Note that for the latter, clusters are scheduled using the reverse order of the top-down scheme.

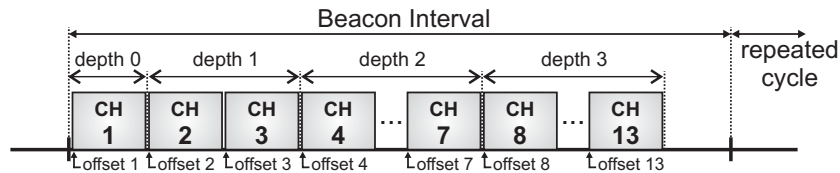


Figure 4.3: Top-down cluster active period scheduling.

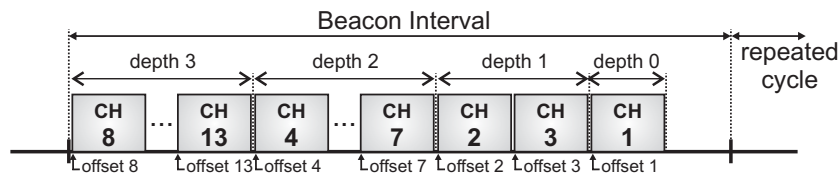


Figure 4.4: Bottom-up cluster active period scheduling.

For the sake of simplicity, no data aggregation or data fusion operations are performed by the cluster-heads. We also assume that the active portions of the clusters are composed only of Contention Access Period (CAP). Finally, we consider that whenever there are error sources affecting the wireless communication, these sources are statistically distributed along the communication environment. Therefore, they equally affect the communication within all the clusters. As the use of a balanced bandwidth allocation scheme is to guarantee a fair distribution of the available communication resources (bandwidth) among all cluster-heads, we do not consider the error behaviour of the communication channel.

For the message traffic model, we assume that sensing nodes periodically send messages to a base station (*sink* node, defined as PAN Coordinator), through the tree path routing from the sensor nodes to the sink node (upstream traffic). Within this context, messages are modelled by a set \mathbb{S} of M message streams:

$$\mathbb{S} = S_1, S_2, \dots, S_M, \quad (4.2)$$

Messages generated by each message stream S_i may be the consequence of regular measurements of environment variables. In this context, a message stream S_i may be characterized as follows:

$$S_i = (C_i, P_i), \quad (4.3)$$

- Where P_i is the period of a cyclic message stream S_i . Message periods are not synchronised with the beacon interval of the network.
- C_i is the length of each message in a message stream S_i . This parameter corresponds to the amount of time required to access to wireless channel, to transmit entirely the message, and to receive the acknowledgement, when required.
- The k -th message of a message streams S_i is represented by M_i^k parameter.

As messages are forwarded through the cluster-tree, the utilisation factor imposed by a specific message stream depends on the depth of its generator node. We define U as the total effective utilisation factor imposed by set \mathbb{S} :

$$U = \sum_{i=1}^M \frac{C_i}{P_i} \times depth_i, \quad (4.4)$$

where $depth_i$ corresponds to the depth of the node that generates S_i (considering the root node to be depth 0, according to Figure 4.2).

4.5 Protocol Constraints

The allocation of superframe durations to each of the clusters is done according to a set of boundary equations, hereafter referred as protocol and buffer constraints. In this section, we model the protocol and buffer constraints used along this work, considering the main requirements imposed by the IEEE 802.15.4 standard.

4.5.1 Length of the Beacon Interval

The length of the beacon interval is an important design parameter for setting-up a cluster-tree network. According to the IEEE 802.15.4 standard, the beacon interval (BI) and the superframe duration (SD_j) are defined as follows:

$$\begin{cases} BI = \alpha \times 2^{BO} \\ SD = \alpha \times 2^{SO}, \text{ for } 0 \leq SO \leq BO \leq 14, \end{cases} \quad (4.5)$$

where α corresponds to the *aBaseSuperframeDuration* MAC parameter. From Equation (4.5), it follows that:

$$\alpha \times 2^0 \leq SD \leq BI \leq \alpha \times 2^{14}, \quad (4.6)$$

where BI and SD values are related to a power of 2^{BO} and 2^{SO} , respectively.

On one hand, the beacon interval must be large enough to ensure that all desired superframe durations can be scheduled, but it should also be as small as possible to reduce the end-to-end delay of message transfers:

$$BI \geq \sum_{j=1}^{N_{CH}} SD_j, \quad (4.7)$$

On the other hand, considering that a message stream S_i generates a new message to be transferred every period P_i , and that any node is able to send messages only during its cluster active period, there is a direct restriction imposed by the beacon interval upon the message stream periodicity.

As the message generation period is not synchronised with the beacon arrivals, it may occur (Figure 4.5) a message M_i^k being generated immediately before the end of the active period of the cluster, so that there is no enough time to transmit this message. As a consequence, this message would only be transmitted during the next active period.

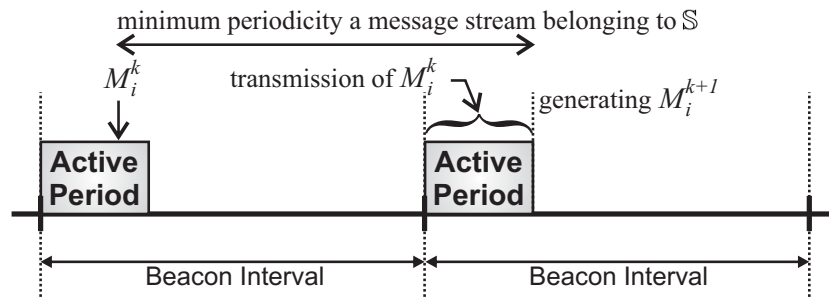


Figure 4.5: Length of beacon interval regarding message periodicities.

Thus, assuming δ as the minimum required time (contention time plus transmission and acknowledgement times) to transmit message M_i^k , in order to guarantee that M_i^k may be transferred before the generation of the next message, period P_i must be, at least, larger than the beacon interval plus δ . Considering the set $\mathbb{P} = \{P_1, P_2, \dots, P_M\}$ as the message periods of set \mathbb{S} of message streams, there is the following constraint upon the beacon interval:

$$BI \leq \min\{\mathbb{P}\} - \delta, \quad (4.8)$$

where $\min\{\mathbb{P}\}$ corresponds to the shortest period for the set \mathbb{S} of defined message streams.

Therefore, from Equations (4.7) and (4.8), a Protocol Constraint that must be satisfied by the cluster-tree network is as follows:

$$\sum_{j=1}^{N_{CH}} SD_j \leq BI \leq \min\{\mathbb{P}\} - \delta, \quad (4.9)$$

Considering that a cluster-tree network is a multihop network, a message to be transferred from source to sink must go through a sequence of clusters, during a sequence of scheduled active periods. That is, the example illustrated in Figure 4.5 must consider the cluster active period schemes illustrated in Figures 4.3 and 4.4.

A consequence of this multihop operating behaviour is that different scheduling approaches may (or not) prioritise the upstream traffic. For example, the constraint defined in Equation 4.8 is adequate for a bottom-up scheduling scheme, as a message generated at the deepest source node is able to reach the sink node during just one beacon interval. However, using a top-down scheduling scheme, the generated message will take several beacon intervals to be delivered to the sink node. Thus, the protocol constraint that must be satisfied by the cluster-tree network is as follows:

$$depth_{MAX} \times BI \leq \min\{\mathbb{P}\} - \delta, \quad \text{for top-down scheduling,} \quad (4.10)$$

Therefore, from Equations (4.9) and (4.10), the **Protocol Constraint** that must be satisfied by the superframe duration of any cluster in a cluster-tree WSN is:

$$\sum_{j=1}^{N_{CH}} SD_j \leq BI \leq \begin{cases} P_{min} - \delta & \text{for bottom-up scheduling} \\ \frac{P_{min} - \delta}{depth_{MAX}} & \text{for top-down scheduling} \end{cases} \quad (4.11)$$

Considering the defined protocol constraint (Equation 4.11) and the MAC constraint imposed by Equation 4.6, the beacon interval can assume just an integer number of values ranging from the sum of superframe durations up to the shortest period of the set of message streams (P_{min}). This consideration represents an important design constraint for cluster-tree networks. Figures 4.6 and 4.7 illustrate this design constraint.

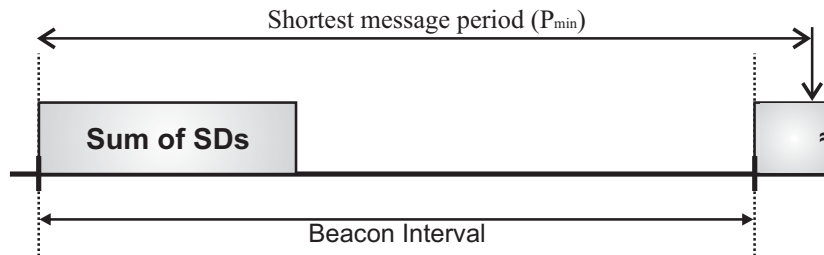


Figure 4.6: Beacon interval set to a value close to the shortest message period.

In Figure 4.6, the beacon interval is set to a value close to the shortest message period (longer beacon interval), enhancing the network lifetime due to the reduction of energy consumption [113]. However, for this configuration, messages streams will have longer end-to-end delays.

On the other hand, Figure 4.7 shows the configuration, where the beacon interval is set to a value close to the sum of superframe durations of the clusters. This configuration represents a shorter beacon interval, that still satisfies the protocol constraint. As a consequence, message streams will have shorter end-to-end delays. However, the energy consumption of nodes will significantly increase.

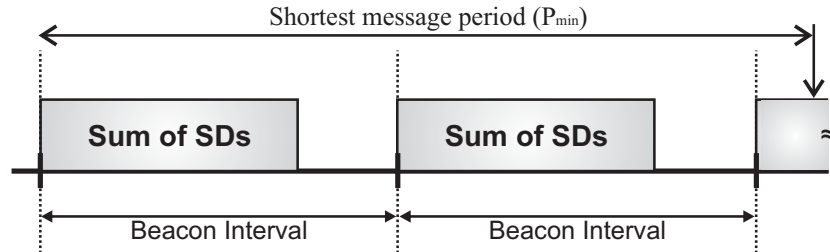


Figure 4.7: Beacon interval set to a value close to the sum of superframe durations.

4.5.2 Dimensioning Buffer Sizes

The correct dimensioning of the MAC buffers for cluster-head nodes is a critical issue, because it has a clear impact upon message discards in cluster-tree networks [99]. Thus, it is important to define a boundary equation for the buffer usage that should be imposed to the overall scheduling of the network. This boundary equation is of major relevance for avoiding message discards due to buffer overflows.

Regardless of any other issue, each cluster-head must be able to store, in the worst case, all messages generated by its descendant nodes (child nodes). That is:

$$\eta \leq \sum_{i \in S_{below}^j} S_i \leq B_j, \quad (4.12)$$

Where B_j is the size of the MAC buffer of cluster-head j (expressed in terms of number of messages), and η is the number of messages generated by the S_{below}^j message streams located in the descendant nodes (child nodes) of CH_j during one beacon interval.

Therefore, we have the following **Buffer Constraint** that must also be satisfied by the buffers at each cluster of the cluster-tree WSN:

$$B_j \geq \sum_{n \in S_{below}^j} S_n, \quad \text{for } 1 \leq n \leq N_{nodes}, \quad (4.13)$$

We assume that messages in the buffers of cluster-heads are ordered by priority (Rate-Monotonic priority scheme [114]), i.e., message streams with the shortest period will have the highest priority.

4.6 Modelling the Transmission Time During the Superframe Duration

As previously highlighted, the network load imposed by a message stream S_i is constrained by the CSMA-CA parameters, such as channel access time and transmission time of a data

frame during active periods. Due to the probabilistic behaviour of the CSMA-CA protocol, we consider the use of a set of communication models proposed by Kohvakka et al. [110] to estimate the frame transmission capacity within a superframe duration, as follows.

Within the active duration of a superframe, the transmission time (T_{TXD}) for a single data frame can be modelled as follows:

$$T_{TXD} = T_{BOT} + T_{PACKET} + T_{radio_transition} + T_{ACK}, \quad (4.14)$$

where T_{BOT} is the total backoff time, T_{PACKET} is the packet transmission time, that is given by $\frac{L_{PACKET}}{D}$ (where L_{PACKET} corresponds to packet frame length and D is the radio data rate). $T_{radio_transition}$ corresponds to time duration that the radio takes to switch between the different operation modes, for example, from sleep to receive modes and from receive to transmit modes. T_{ACK} corresponds to the acknowledgement transmission time, that is given by $\frac{L_{ACK}}{D}$ (where L_{ACK} corresponds to the ack frame length). Figure 4.8 illustrates the basic scheme for transmission of a single data frame. Note that the backoff period is aligned with the beacon period and nodes must perform the backoff algorithm to transmit a data frame.

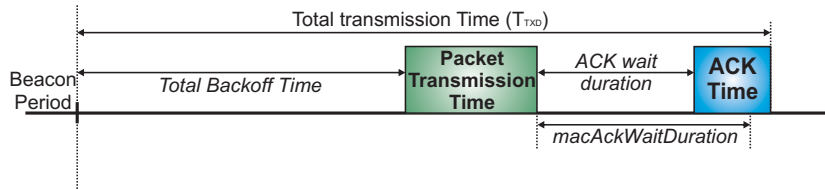


Figure 4.8: Transmission time duration of a data frame.

According to the CSMA-CA algorithm, a node needs to perform two CCA before transmitting a packet (CW_{init} parameter default is 2). For this, Kohvakka et al. [110] model the probability (\mathbb{P}_c) of a node to perform two consecutive CCA as follows:

$$\mathbb{P}_c = (1 - q)^{2 \times (N_{nodes} - 1)}, \quad (4.15)$$

where q corresponds to the probability that a node transmits a single message (with its ack) at any time of the CAP and can be modelled as [110]:

$$q = \frac{L_{PACKET}}{SD \times R}, \quad (4.16)$$

The CSMA-CA algorithm also defines a maximum number of backoffs (b), which correspond to the number of attempts that the backoff algorithm is repeated in case of unsuccessfully CCA attempt. This value is defined by $macMaxBackoffPeriod$ MAC parameter and its default value is 4. Hence, the probability (\mathbb{P}_s) of a node to perform a CCA (access to channel) with the maximum backoff number (b) can be modelled as:

$$\mathbb{P}_s = \sum_{NB=1}^b P_c \times (1 - P_c)^{NB-1}, \quad (4.17)$$

With this, the average backoff number (r) for each message can be modelled as [110]:

$$r = (1 - P_s) \times b + \sum_{NB=1}^b NB \times P_c \times (1 - P_c)^{NB-1}, \quad (4.18)$$

Also, the average backoff time (T_{BO}) for each message can be modelled as function of backoff exponential (BE) [110]:

$$T_{BO}(BE) = \frac{2^{BE} - 1}{2} \times T_{BOL}, \quad (4.19)$$

where T_{BOL} corresponds to the backoff period length and its value is defined by *aUnitBackoffPeriod* MAC parameter (default value is 20 symbols).

Therefore, the total backoff time can be modelled based on Equations 4.18 and 4.19 and considering that, for each backoff period, in average 3/2 CCA analysis are performed [110]. Thus, the total backoff time (T_{BOT}) can be obtained as follows:

$$T_{BOT} = \frac{3}{2}r(T_{CCA}) + \sum_{NB=0}^{r-1} T_{BO}(\min(\text{macMinBE} + NB, \text{macMaxBE})), \quad (4.20)$$

where T_{CCA} corresponds to the CCA analysis time.

These models are considered in this work in order to predict the number of messages that can be transferred during a minimum superframe duration.

Finally, the collision probability (\mathbb{P}_{co}) of two nodes can be modelled as:

$$\mathbb{P}_{co} = \frac{1}{2^{BE} - 1}, \quad (4.21)$$

4.7 Timing Constraints

In this section, we derive a timing boundary equation associated to the transfer of a message, considering the set of message streams defined for the cluster-tree network and the probabilistic behaviour of the CSMA-CA algorithm modelled in previous section. The main target of this boundary equation is to enable the network designers to predict the probabilistic timing behaviour associated to the message streams and to adjust, case necessary, the periodic message stream model or the protocol parameters of the cluster-tree network. It is important to note that, as cluster-tree monitoring networks basically deal with softly-bounded message delays, this analysis provides a probabilistic methodology to avoid message drops due to buffer overflows, the congestion of the network and predictable end-to-end delays associated to each of the message streams.

The target is to define a boundary equation associated to the end-to-end delay for transferring data frames, considering the inherent characteristics of cluster-tree networks. This boundary equation will then be compared with the periodicity of each message stream,

in order to assess its (probabilistic) schedulability. The work presented in this paper considers earlier work from Lange et al. [104] and Agrawal et al. [115], which use similar approaches for the schedulability analysis of FlexRay and FDDI networks, respectively. The response time analysis calculations are based on earlier work presented by Audsley et al. [116], for the response time analysis of multi-task scheduling on mono-processors.

For computing a boundary equation for the response time of a message stream, we consider a worst-case scenario. In this scenario, all messages streams simultaneously generate messages before the end of cluster active period in the first beacon interval, but it does not have enough time to transmit any of these messages. In this case, all messages will be queued in the internal buffers of the nodes. Figure 4.9 illustrates this case, in which the transmissions will only start in next active period scheduling, each one during its active period.

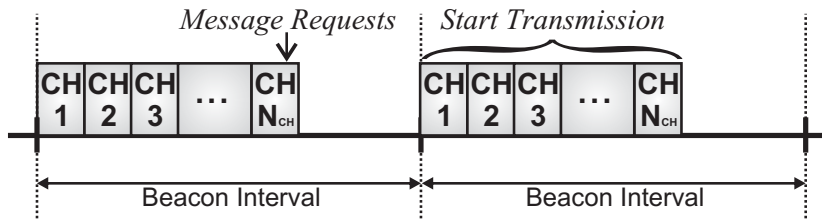


Figure 4.9: Worst-case scenario for cluster-tree networks.

In order to model a boundary equation for the response time (R_i) of a message stream S_i , we need to derive its local worst-case response time (W_i), which corresponds to the Worst-Case Response Time (WCRT) for a specific cluster of any depth. Within this context, we assume the worst-case scenario for message M_i^k of message stream S_i , according to Figure 4.10.

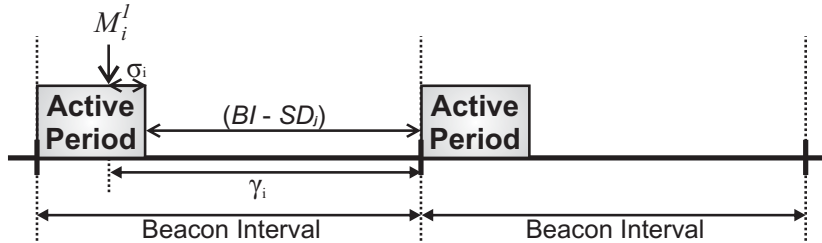


Figure 4.10: Local worst-case response time for a message stream.

Note that, in the best-case scenario, this message will be transmitted to its parent cluster-head during the next cluster active period. This way, we consider an initial delay γ_i in which a message must wait by the next active period of its cluster. This initial delay can be expressed as follows:

$$\gamma_i = \sigma_i + (BI - SD_j), \quad (4.22)$$

where σ_i corresponds to a time interval immediately smaller than the transmission time for one message M_i^k within the CAP and $(BI - SD_j)$ corresponds to the inactive period for that cluster.

However, this message can suffer interferences from the set of higher priority message streams, which will impose extra delays for this message. Thus, for message M_i^k , we model

the interference (Θ_i) imposed by the higher priority message streams located in the descendant nodes of cluster-head CH_j of the current active cluster j . In the best-case scenario, this interference corresponds to the sum of the transmission times of the set of higher priority message streams. However, if this sum exceeds the superframe duration, the subsequent superframe durations must also be considered. Thus, we define (Θ_i) for message M_i^k as:

$$\Theta_i^1(CH_j) = T_{TXD_i} + \left\lfloor \frac{\sum_{\substack{h \in \mathbb{S}_{below}^j \\ h \in hp(i)}} T_{TXD_h}}{SD_j} \right\rfloor \times (BI - SD_j) + \sum_{\substack{h \in \mathbb{S}_{below}^j \\ h \in hp(i)}} T_{TXD_h} \quad (4.23)$$

$$\begin{aligned} \Theta_i^w(CH_j) = T_{TXD_i} &+ \left\lfloor \frac{\sum_{\substack{h \in \mathbb{S}_{below}^j \\ h \in hp(i)}} \left\lceil \frac{\Theta_i^{w-1}(j)}{P_h} \right\rceil \times T_{TXD_h}}{SD_j} \right\rfloor \times (BI - SD_j) \dots \\ &\dots + \sum_{\substack{h \in \mathbb{S}_{below}^j \\ h \in hp(i)}} \left\lceil \frac{\Theta_i^{w-1}(j)}{P_h} \right\rceil \times T_{TXD_h}, \quad \text{for } w > 1 \end{aligned} \quad (4.24)$$

Note that the interference Θ_i is modelled as function of the active period of cluster-head CH_j , because message M_i^k suffers interferences of multiple message streams \mathbb{S}_{below}^j every time it is forwarded along the cluster-tree path. The w iterations are performed until $\Theta_i^w = \Theta_i^{w-1}$.

Based on the interference time imposed by the highest priority message load, we derive a local worst-case response time (W_i) for message M_i^k as:

$$W_i = \gamma_i + \Theta_i^w, \quad (4.25)$$

In addition, the message must traverse all the cluster-tree path from the source cluster until the PAN coordinator. Thus, the transmission delay towards the sink node takes into account the depth of the node that generated the message stream. Within this context, we derive the total probabilistic worst-case response time (R_i) for message M_i^k as function of the network depth (regarding to the PAN coordinator) of the generation node of this message stream as follows:

$$R_i = \gamma_i + \sum_{d=depth_n-1}^0 \Theta_i^w(CH_j(d)), \quad (4.26)$$

where $CH(d)$ corresponds to cluster-head CH_j of depth d that is responsible for forwarding the message M_i^k along the cluster-tree path.

Moreover, the worst-case response time is also dependent of the active period scheduling scheme. For this reason, it is added an additional delay that is dependent of the used scheduling scheme. For the bottom-up scheduling scheme, where the upstream traffic is prioritised, a worst-case delay must be added that corresponds to the sum of active periods of all clusters. In turn, for the top-down scheduling scheme, where the message

needs to use multiple beacon intervals before reaching the PAN coordinator, we consider a pessimistic worst-case delay that corresponds to the difference between one beacon interval and the superframe duration of the responsible cluster for forwarding the message at each depth of the cluster-tree network until reaching the PAN coordinator. These assumptions are appropriate, because they encompass worst-case scenarios for the scheduling schemes. Therefore, the worst-case response time (R_i) can be derived as follows:

$$R_i = \begin{cases} \gamma_i + \sum_{d=\text{depth}_n-1}^0 \Theta_i^w(CH_j(d)) + \sum_{j=1}^{N_{CH}} SD_j, & \text{bottom-up} \\ \gamma_i + \sum_{d=\text{depth}_n-1}^0 \Theta_i^w(CH_j(d)) + \sum_{d=\text{depth}_n-1}^0 (BI - SD_{CH_j(d)}), & \text{top-down} \end{cases} \quad (4.27)$$

Within this context, when specifying the probabilistic worst-case response time (R_i) for message M_i^k , a boundary equation (**timing constraint**) that must be satisfied by the cluster-tree network scheduling can be defined as follows:

$$R_i \leq P_i, \quad \text{for } \forall S_i \in \mathbb{S}, \quad (4.28)$$

4.8 Superframe Duration Allocation Schemes

Finally, we derive a set of Superframe Duration Allocation (SDA) schemes intended to improve the throughput of the monitoring traffic in cluster-tree networks. The underlying reasoning is to estimate adequate superframe duration values (SD) and buffer sizes for each cluster coordinator, considering both the network requirements and the protocol and timing constraints. These allocation schemes can help the designer of cluster-tree networks in what concerns the definition of network parameters, configuration of message streams and the need of using techniques such as data fusion or aggregation.

Within this context, we propose two proportional allocation schemes for setting-up the superframe durations: 1) Load-SDA, based on the traffic load imposed by the cluster descendant nodes, and 2) Nodes-SDA, based on the number of descendant nodes. The Load-SDA scheme is suitable for cluster-tree networks where both the topology and the data traffic behaviour are known. In turn, the Nodes-SDA scheme is suitable for cluster-tree networks, where only the topology is known. In this case, the Nodes-SDA allocation scheme assumes a similar traffic load generated by each sensor node.

The use of proportional allocation schemes to define the superframe duration of each cluster-head improves the throughput of the cluster-tree network, as it ensures adequate network resources (bandwidth) to each cluster-head to transmit message streams according to their requirements. Therefore, the use of this type of allocation schemes can avoid network congestion and message discards due to buffer overflows, that usually occur near the PAN coordinator [99]. Moreover, by defining adequate bandwidth and buffer sizes for cluster coordinators, the proposed schemes may also guarantee a minimum level of quality-of-service (QoS) for the message streams, and a smaller energy consumption level for each of the network cluster-heads.

4.8.1 Proportional Allocation Scheme based on the Message Load

In this subsection, we define a *Proportional Superframe Duration Allocation* scheme, that allocates bandwidth to a specific cluster based on its message load (**Load-SDA**). The reasoning is to proportionally allocate superframe durations based on the message traffic of each cluster and their child nodes, including the accumulated message traffic of child coordinators. This scheme considers that both the network topology and the data traffic behaviour are known. The Load-SDA scheme is described in Algorithm 1.

Algorithm 1: Load-SDA Algorithm.

```

1 // Step 1: define the beacon Interval (BI)
2 if (Scheduling Scheme == Bottom-Up) then
3   |  $BI \leq P_{min} + \delta_i$ ;
4 else
5   |  $BI \leq \frac{P_{min} + \delta_i}{depth_{MAX}}$ ;
6 // Step 2: define the capacity  $X$  of the  $SD_{min}$ 
7  $X$  = number of messages transferred during  $SD_{min}$  (Equation 4.29);
8 // Step 3: define the Superframe Duration for each cluster
9 for  $j := 1$  to  $N_{CH}$  do
10  | Compute  $Y$  for  $CH_j$  based on Equation 4.30;
11  | Compute  $SD_j$  for  $CH_j$  based on Equation 4.33;
12 // Step 4: test the Protocol Constraint
13 Compute the Protocol Constraint based on Equation 4.11;
14 if (Protocol Constraint is not satisfied) then
15  | System is not schedulable;
16 // Step 5: test the Timing Constraint
17 for  $j := 1$  to  $m$  do
18  | Compute the Timing Constraint based on Equations 4.27 and 4.28;
19  | if (Timing Constraint is not satisfied) then
20    | | Message Stream Configuration or Network Parameters are not adequate for the
    | | System;

```

In the first step, the Load-SDA algorithm defines a value for the beacon interval (BI) considering the constraint imposed by Equation 4.11. We model the number X of messages transferred during the minimum superframe duration SD_{min} (step 2 in Algorithm 1) as follows:

$$X = \left\lfloor \left(\frac{SD_{min}}{t_{TXD}} \right) \times (1 - \mathbb{P}_{co})^m \right\rfloor, \quad (4.29)$$

where SD_{min} corresponds to the SO parameters equal to 0 in Equation 4.6, t_{TXD} correspond to the total transmission time for a single message, and m represents the number of communicating nodes within a cluster times the probability q of a node to transmit a message at any time. For this analysis, we consider the number of communicating nodes within a cluster as the maximum number of nodes per cluster, which is a parameter defined before the formation phase of the cluster-tree network.

Thus, to define the SD_j for cluster-head CH_j , the Load-SDA algorithm considers the load

Y imposed by all message streams hierarchically below the analysed cluster-head (step 3 in Algorithm 1, line 12), including its child nodes and the accumulated message load imposed by each child cluster-head, which is modelled as:

$$Y = \sum_{i \in \mathbb{S}_{below}} \frac{1}{\left\lceil \frac{P_i}{BI} \right\rceil}, \quad (4.30)$$

where $\left\lceil \frac{P_i}{BI} \right\rceil$ corresponds to the maximum number of messages generated by S_i during BI .

From Equations 4.29 and 4.30, the algorithm defines the necessary number of SD_{min} to guarantee the load imposed by the message streams. Thus, we have:

$$SD_j = \left\lceil \frac{Y}{X} \right\rceil \times SD_{min}, \quad (4.31)$$

Following the constraints imposed for Equation 4.6, it follows that:

$$SD_j = \alpha \times 2^{SO_j}, \quad (4.32)$$

From Equations 4.31 and 4.32, to define the superframe duration (SD_j) for cluster-head CH_j (step 3 in Algorithm 1, line 13), we derive the following equation to calculate its superframe order (SO_j):

$$\begin{aligned} \alpha \times 2^{SO_j} &= \left\lceil \frac{Y}{X} \right\rceil \times \alpha \times 2^0 \\ SO_j &= \left\lceil \log_2 \left(\left\lceil \frac{Y}{X} \right\rceil \right) \right\rceil \end{aligned} \quad (4.33)$$

After allocating a superframe duration for all cluster-heads, the protocol and timing constraints (Equations 4.11 and 4.28) must be verified (steps 4 and 5 in Algorithm 1, lines 15-25). Case the protocol or the timing constraints are not satisfied, it means that the system may not be schedulable and it is necessary to modify the configuration of the network and/or the set of supported messages streams. Unfortunately, the IEEE 802.15.4 standard provides a reduced flexibility to modify the values for the Superframe Duration (SD) and Beacon Interval (BI) parameters. The main reason is that these parameters are described by the Superframe Order (SO) and Beacon Order (BO) parameters, which are related with each other by a power of two. Thus, any adjust of SO or BO can significantly change the values of SD and BI, respectively.

4.8.2 Proportional Allocation Scheme Based on the Number of Nodes

Within the same context, we also propose a *proportional Superframe Duration Allocation* scheme based on the number of nodes (**Nodes-SDA**), which proportionally allocates a Superframe Duration (SD) for each cluster based on the number of its descendant nodes, without considering the load imposed by each one of them. Differently to the Load-SDA scheme, this scheme is suitable for applications where the load imposed by each cluster is unknown. The Nodes-SDA scheme is described in Algorithm 2.

Algorithm 2: Nodes-SDA Algorithm.

```

1 // Step 1:  define the beacon Interval ( $BI$ )
2 if ( $Scheduling\ Scheme == Bottom-Up$ ) then
3    $BI \leq P_{min} + \delta_i$ ;
4 else
5    $BI \leq \frac{P_{min} + \delta_i}{depth_{MAX}}$ ;
6 // Step 2:  define the capacity  $X$  of the  $SD_{min}$ 
7  $X =$  number of messages transferred during  $SD_{min}$  (Equation 4.29);
8 // Step 3:  define the Superframe Duration for each cluster
9 for  $j := 1$  to  $N_{CH}$  do
10   Compute  $N$  for  $CH_j$  based on Equation 4.34;
11   Compute  $SD_j$  for  $CH_j$  based on Equation 4.37;
12 // Step 4:  test the Protocol Constraint
13 Compute the Protocol Constraint based on Equation 4.11;
14 if ( $Protocol\ Constraint\ is\ not\ satisfied$ ) then
15   System is not schedulable;
16 // Step 5:  test the Timing Constraint
17 for  $j := 1$  to  $m$  do
18   Compute the Timing Constraint based on Equations 4.27 and 4.28;
19   if ( $Timing\ Constraint\ is\ not\ satisfied$ ) then
20     Message Stream Configuration or Network Parameters are not adequate for the
      System;

```

The first and second steps of the Nodes-SDA algorithm are similar to Load-SDA algorithm. The beacon Interval and the number X of messages transferred during the minimum superframe duration SD_{min} are defined according to Equations 4.11 and 4.29 (lines 3-9 in Algorithm 2). Thus, to define the SD_j for cluster-head CH_j , the Nodes-SDA algorithm just considers the number N of hierarchically descendant nodes below the analysed cluster-head (step 3 in Algorithm 1, line 12), including its child nodes and the accumulated child nodes of each child cluster-head:

$$N = \sum_{S_i \in \mathbb{S}_{below}} 1, \quad \text{for } 1 \leq i \leq M, \quad (4.34)$$

From Equations 4.29 and 4.34, the algorithm defines the necessary number of SD_{min} to guarantee the message stream traffic. Thus, we have:

$$SD_j = \left\lceil \frac{N}{X} \right\rceil \times SD_{min}, \quad (4.35)$$

Following the constraints imposed by Equation 4.6, it follows that:

$$SD_j = \alpha \times 2^{SO_j}, \quad (4.36)$$

From Equations 4.35 and 4.36, to define the superframe duration (SD_j) for cluster-head CH_j (step 3 in Algorithm 2, line 13), we derive the following equation to calculate its respective superframe order (SO_j):

$$\begin{aligned}\alpha \times 2^{SO_j} &= \left\lceil \frac{N}{X} \right\rceil \times \alpha \times 2^0 \\ SO_j &= \left\lceil \log_2 \left(\left\lceil \frac{N}{X} \right\rceil \right) \right\rceil\end{aligned}\quad (4.37)$$

Finally, both the protocol and timing constraints (Equations 4.11 and 4.28) must be verified (steps 4 and 5 in Algorithm 1, lines 15-25).

As the Nodes-SDA scheme allocates superframe durations based on the number of nodes in the network topology, it may over-allocate durations for cluster-heads with lower message loads. In this way, Nodes-SDA commonly generates a larger sum of superframe durations, when compared to Load-SDA.

4.8.3 Illustrative Example

In this subsection, we present an example to illustrate the use of the proposed superframe duration allocation schemes. For this, we consider a small example of a cluster-tree network composed of 6 cluster-heads and each cluster composed of 2 leaf nodes (Figure 4.11). For this example, we only consider the Load-SDA scheme, considering that the topology and message traffic load are known. For the sake of simplicity, all values represented in this subsection will be multiples of SD_{min} . We also consider that each leaf node generates one message stream; nodes with odd indexes generate message streams with periods equal to $60 \times SD_{min}$ and nodes with even indexes generate message streams with periods equal to $70 \times SD_{min}$. In this example, cluster-heads do not generate any message stream.

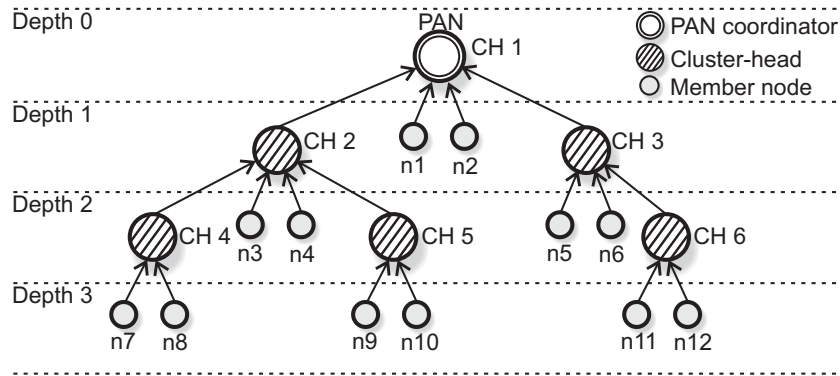


Figure 4.11: Cluster-tree network composed of 6 cluster-heads and 12 leaf nodes.

Based on the analysis presented in Section 4.6, we assume parameter X to be 2 messages transferred during a SD_{min} (Equations 4.14 and 4.29). For this example, we consider a bottom-up scheduling and we apply the Load-SDA scheme. Within this context, the Beacon Interval (BI) is set according to Equation 4.8. Considering that Pi_{min} is $60 \times SD_{min}$, the Beacon Interval (BI) is defined to be $32 \times SD_{min}$ (BO = 5).

Applying the Load-SDA algorithm, the superframe duration is calculated for each cluster-head following a bottom-up direction. Thus, for cluster-head CH_6 , the load imposed by its message streams is (Equation 4.30):

$$Y = \sum_{i \in \mathbb{S}_{below}} \frac{1}{\left\lceil \frac{P_i}{BI} \right\rceil} = \frac{1}{\left\lceil \frac{60}{32} \right\rceil} + \frac{1}{\left\lceil \frac{70}{32} \right\rceil} = 1 + \frac{1}{2} = \frac{3}{2}, \quad (4.38)$$

Therefore, for CH_6 , the superframe order SO_6 is:

$$SO_6 = \left\lceil \log_2 \left(\left\lceil \frac{Y}{X} \right\rceil \right) \right\rceil = \left\lceil \log_2 \left(\left\lceil \frac{3}{4} \right\rceil \right) \right\rceil = 0, \quad (4.39)$$

With this, for CH_6 , the superframe duration is $SD_6 = \alpha \times 2^0$, that corresponds to SD_{min} . In fact, as cluster-head 6 has only two leaf nodes and a SD_{min} supports two messages, the SD provides a reasonable bandwidth for this traffic. Following the same reasoning, superframe durations of cluster-heads 4 and 5 are also defined to SD_{min} .

For cluster-head CH_3 , the message load includes the load imposed by its child nodes (Equation 4.38) and the child cluster-head CH_6 (accumulated load). The accumulated load is:

$$Y = \sum_{i \in \mathbb{S}_{below}} \frac{1}{\left\lfloor \frac{P_i}{BI} \right\rfloor} = 2 \times \frac{1}{\left\lfloor \frac{60}{32} \right\rfloor} + 2 \times \frac{1}{\left\lfloor \frac{70}{32} \right\rfloor} = 2 + 1 = 3, \quad (4.40)$$

Thus, for CH_3 , the superframe order SO_3 is:

$$SO_3 = \left\lceil \log_2 \left(\left\lceil \frac{Y}{X} \right\rceil \right) \right\rceil = \left\lceil \log_2 \left(\left\lceil \frac{3}{2} \right\rceil \right) \right\rceil = 1, \quad (4.41)$$

With this, for CH_3 , the superframe duration is $SD_3 = \alpha \times 2^1$, which corresponds to $2 \times SD_{min}$. Table 4.1 shows the superframe duration for all clusters.

Table 4.1: Superframe Durations for all Clusters.

	SO_i	SD_i
CH_1	3	$8 \times SD_{min}$
CH_2	2	$4 \times SD_{min}$
CH_3	1	$2 \times SD_{min}$
CH_4	0	SD_{min}
CH_5	0	SD_{min}
CH_6	0	SD_{min}

After defining the superframe duration of all cluster-heads, the protocol and timing constraints need be verified. Based on Equation 4.11, the protocol constraint is respected, as follows:

$$\sum_{j=1}^N SD_j \leq BI \leq P_{min} + \delta_i \quad (4.42)$$

$$17 \times SD_{min} \leq 32 \times SD_{min} \leq 60 \times SD_{min} + \frac{1}{2} \times SD_{min} \quad (4.43)$$

$$17 \leq 32 \leq 60 + \frac{1}{2} \quad (4.44)$$

Regarding to the timing constraint, as an example, we present the response time analysis for message stream S_{10} (depth 3). For this work, we assume that message streams with the

same period of the message S_{10} are considered for the set of the highest priority message streams. Thus, for depth 3 (CH_5), the set of higher priority message streams is composed only of message stream S_9 . Thus, we have:

$$\Theta_{10}^1(CH_5) = \frac{1}{2} + \left\lfloor \frac{\frac{1}{2}}{1} \right\rfloor \times 31 + \frac{1}{2} = 1 \quad (4.45)$$

$$\Theta_{10}^2(CH_5) = \frac{1}{2} + \left\lfloor \frac{\left\lceil \frac{1}{60} \right\rceil \times \frac{1}{2}}{1} \right\rfloor \times 31 + \left\lceil \frac{1}{60} \right\rceil \times \frac{1}{2} = 1 \quad (4.46)$$

For depth 2 (CH_2), the set of higher priority message streams is composed of message streams S_3, S_4, S_7, S_8 and S_9 . Thus, we have:

$$\Theta_{10}^1(CH_2) = \frac{1}{2} + \left\lfloor \frac{\frac{1}{2} \times 5}{4} \right\rfloor \times 28 + \frac{1}{2} \times 5 = 3 \quad (4.47)$$

$$\Theta_{10}^2(CH_2) = \frac{1}{2} + \left\lfloor \frac{\left\lceil \frac{3}{60} \right\rceil \times \frac{1}{2} \times 3 + \left\lceil \frac{2}{70} \right\rceil \times \frac{1}{2} \times 2}{4} \right\rfloor \times 28 + \left\lceil \frac{3}{60} \right\rceil \times \frac{1}{2} \times 3 + \left\lceil \frac{2}{70} \right\rceil \times \frac{1}{2} \times 2 = 3 \quad (4.48)$$

In turn, for depth 1 (CH_1), the set of higher priority message streams is composed of message streams $S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{11}$ and S_{12} . Thus, we have:

$$\Theta_{10}^1(CH_1) = \frac{1}{2} + \left\lfloor \frac{\frac{1}{2} \times 11}{8} \right\rfloor \times 24 + \frac{1}{2} \times 11 = 6 \quad (4.49)$$

$$\Theta_{10}^2(CH_1) = \frac{1}{2} + \left\lfloor \frac{\left\lceil \frac{6}{60} \right\rceil \times \frac{1}{2} \times 6 + \left\lceil \frac{6}{70} \right\rceil \times \frac{1}{2} \times 5}{8} \right\rfloor \times 24 + \left\lceil \frac{6}{60} \right\rceil \times \frac{1}{2} \times 6 + \left\lceil \frac{6}{70} \right\rceil \times \frac{1}{2} \times 5 = 6 \quad (4.50)$$

Finally, applying Equation 4.27 for the bottom-up scheduling, we have:

$$R_i = \sum_{j=1}^{N_{CH}} SD_j + \gamma_i + \sum_{d=1}^{depth_n} \Theta_i^w(CH_j(d)) \quad (4.51)$$

$$R_{10} = \sum_{j=1}^{N_{CH}} SD_j + \sigma_{10} + (BI - SD_5) + \Theta_{10}^w(CH_5) + \Theta_{10}^w(CH_2) + \Theta_{10}^w(CH_1) \quad (4.52)$$

$$R_{10} = 17 + \frac{1}{2} + (32 - 1) + 1 + 3 + 6 = 58, 5 \quad (4.53)$$

considering σ_{10} as approximately $\frac{1}{2} \times SD_{min}$. Thus, the timing constraint for the message stream S_{10} is satisfied, as follows:

$$R_{10} \leq P_{10} \quad (4.54)$$

$$58, 5 \leq 70 \quad (4.55)$$

Table 4.2 shows the timing constraints for all message streams, from where it is clear that all deadlines can be met (a probabilistic guarantee, as the underlying message duration is probabilistic).

The purpose of this example is just to illustrate the use of allocation scheme Load-SDA. Noticeably, for a real cluster-tree network, this analysis must be extended to the overall set of cluster-heads.

Table 4.2: Timing Constraint for all Message Streams.

S_i	R_i	P_i	$R_i \leq P_i$
S_1	$44,5 \times SD_{min}$	$60 \times SD_{min}$	ok
S_2	$47,5 \times SD_{min}$	$70 \times SD_{min}$	ok
S_3	$50 \times SD_{min}$	$60 \times SD_{min}$	ok
S_4	$54,5 \times SD_{min}$	$70 \times SD_{min}$	ok
S_5	$51,5 \times SD_{min}$	$60 \times SD_{min}$	ok
S_6	$54,5 \times SD_{min}$	$70 \times SD_{min}$	ok
S_7	$53,5 \times SD_{min}$	$60 \times SD_{min}$	ok
S_8	$58,5 \times SD_{min}$	$70 \times SD_{min}$	ok
S_9	$53,5 \times SD_{min}$	$60 \times SD_{min}$	ok
S_{10}	$58,5 \times SD_{min}$	$70 \times SD_{min}$	ok
S_{11}	$53 \times SD_{min}$	$60 \times SD_{min}$	ok
S_{12}	$56,5 \times SD_{min}$	$70 \times SD_{min}$	ok

4.9 Simulation Assessment

Finally in this section, it is presented a simulation assessment of the allocation schemes proposed in this paper. The main objective is to analyse the network behaviour when applying the proposed allocation schemes and to compare it with the case of a cluster-tree network configured according to IEEE 802.15.4 standard.

For this simulation assessment, we have implemented a simulation model for cluster-tree networks using the Castalia Simulator [117]. Castalia¹ is an open-source discrete event simulator for WSNs, Body Area Networks (BAN) and general low-power embedded networks, that was developed at National ICT Australia (NICTA) and is based on the OMNeT++ platform. Castalia has become a very popular simulator, widely used by researchers and developers to test their protocols using realistic wireless channel and radio models [87]. Castalia implements an advanced wireless channel model, based on empirically measured data. Also, the simulator provides radio models based on real low-power communication radios. Moreover, important features to simulate WSNs are available, such as: realistic node behaviour, node *clock drift*, and energy consumption models.

Castalia provides an IEEE 802.15.4 model. However, this model is quite limited. Basically, it only implements the CSMA-CA functionality and a beacon-enabled star topology, including an association procedure, direct data transfer mode, and GTS communication. Therefore, we developed a simulation model, which includes a series of multi-hop functionalities, such as: cluster-tree formation procedure, network scheduling, hierarchical addressing scheme, direct and indirect data communication, collision domain definition, data communication to the sink node (PAN coordinator), and the proposed superframe duration allocation schemes.

4.9.1 Simulation Environment

For this simulation assessment, a communication environment was defined with size of $200\text{ m} \times 200\text{ m}$, composed of 201 sensor nodes (one PAN coordinator, plus 200 sensing nodes).

¹The Castalia Simulator for Wireless Sensor Network: <https://castalia.forge.nicta.com.au>.

The PAN coordinator was located in position $5\text{ m} \times 5\text{ m}$ of the environment, while 200 sensing nodes were randomly deployed. The PAN coordinator node was deployed in the corner of the environment, in order to build deep cluster-tree networks. Figure 4.12 illustrates an example of the simulation environment used in this assessment.

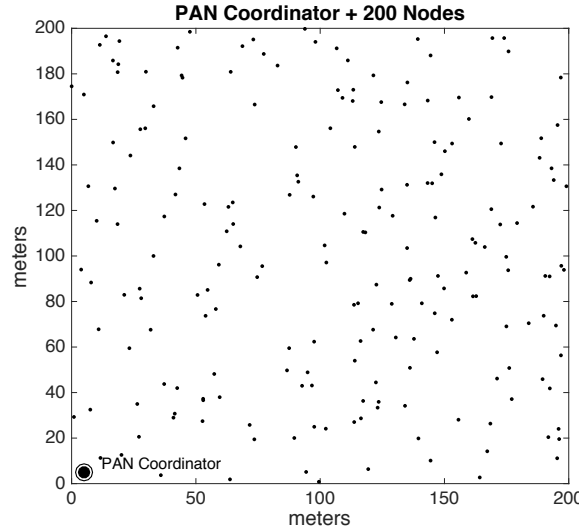


Figure 4.12: Configuration of environment with one PAN coordinator and 200 randomly deployed nodes.

Regarding to the monitoring traffic, sensing nodes generate periodic message streams and send them to sink node. For the sake of simplification, we defined that each sensing node only generates one message stream and the PAN coordinator is the sink node for all monitoring traffic (it does not generate any traffic). Each sensing node generates 1.000 data messages and sends them to the PAN coordinator, following the rules defined by IEEE 802.15.4/ZigBee data communication. Thus, data messages are forwarded along the cluster-tree network according to the tree routing. Importantly, the cluster-heads do not perform any data aggregation or fusion mechanism, which implies that all monitoring traffic is forwarded towards the sink node. In order to generate different message loads for the cluster-heads, we define two different data rates for the set of message streams: a higher data rate (0.05 pkts/s – periodicity of 20 *seconds*), and a lower data rate (0.01 pkts/s – periodicity of 100 *seconds*).

The cluster-tree formation process is based on the IEEE 802.15.4/ZigBee standards. The PAN coordinator (defined as the depth 0 of the cluster-tree network) is responsible to trigger the formation process by building its own cluster and acting as cluster-head. We defined the maximum number of child nodes per cluster to be 6 (six) nodes. Within this context, for this simulation assessment, we have defined two cluster-tree formation procedures, in order to create two different simulation scenarios: an unconditioned cluster-tree formation (hereafter called *unconditioned Scenario*) and a conditioned cluster-tree formation (hereafter called *conditioned Scenario*).

In the first scenario (unconditioned formation), each CH (including the PAN coordinator) can select a maximum number of 2 (two) candidate child nodes to be cluster-heads. The

selection of CH candidates is randomly performed. With this, the cluster-tree network can grow in any direction. Each CH candidate can build its own cluster, following the same rules. The data rates are randomly and evenly distributed along the sensing nodes in the network environment. In this case, we have a randomly distributed message load for the cluster-tree network. Figure 4.13 shows an example of the *unconditioned Scenario*. Figure 4.13(a) illustrates the data rates randomly distributed along the environment, while Figure 4.13(b) illustrates an example of the physical topology for the *unconditioned Scenario*.

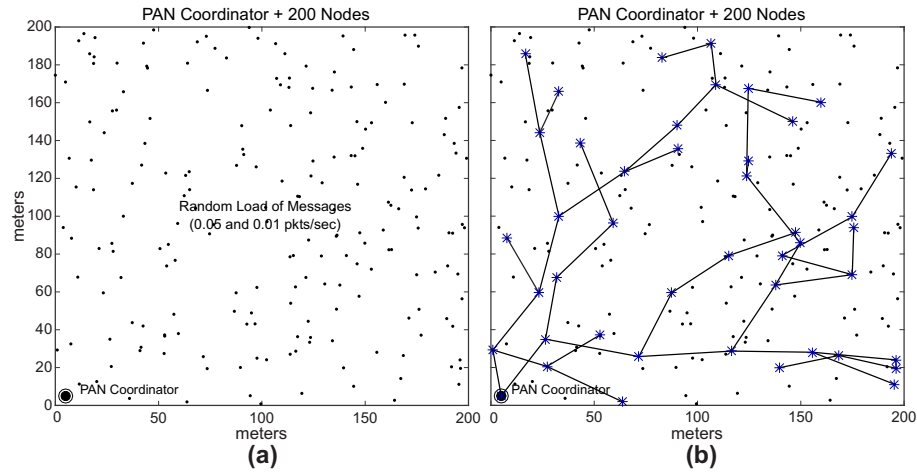


Figure 4.13: Configuration of the *unconditioned Scenario*: (a) Randomly distributed data rates; (b) Unconditioned cluster-tree network.

In the second scenario (conditioned formation), we have equally divided the environment in two different load zones: a high load zone, and a low load zone. With this, the nodes located in the high load zone are configured with data rate of 0.05 pkts/s (higher data rate). In turn, nodes located in the low load zone are configured with data rate of 0.01 pkts/s (lower data rate).

Considering these two different load zones, the cluster-tree formation process is started by the PAN coordinator, which is responsible for building its own cluster. In order to build a conditioned cluster-tree network, PAN coordinator selects one CH candidate located in the high load zone and another candidate located in the low load zone. Following, each cluster-head can select a maximum number of 3 (three) CH candidates, that must also be located in the same load zone of their parent CHs. Therefore, we have a conditioned cluster-tree network, where one branch is built along the high load zone and the other branch is built along the low load zone. Figure 4.14 shows this *conditioned Scenario*. Figure 4.14(a) illustrates the two defined load zones in the environment (high and low load zones), while Figure 4.14(b) illustrates an example of the physical topology for the *conditioned Scenario*.

Table 4.3 summarises the main features of the *unconditioned* and *conditioned Scenarios*. The rationale behind these two scenarios is the following: the first scenario enables a direct comparison against the standard IEEE 802.15.4 configuration (*unconditioned Scenario*); the second scenario enables the comparison between both proportional allocation schemes (*conditioned Scenario*).

For this simulation assessment, we used the ZigBee-based *hierarchical addressing* scheme, in which each CH has its own sequential address block. Regarding to the cluster's

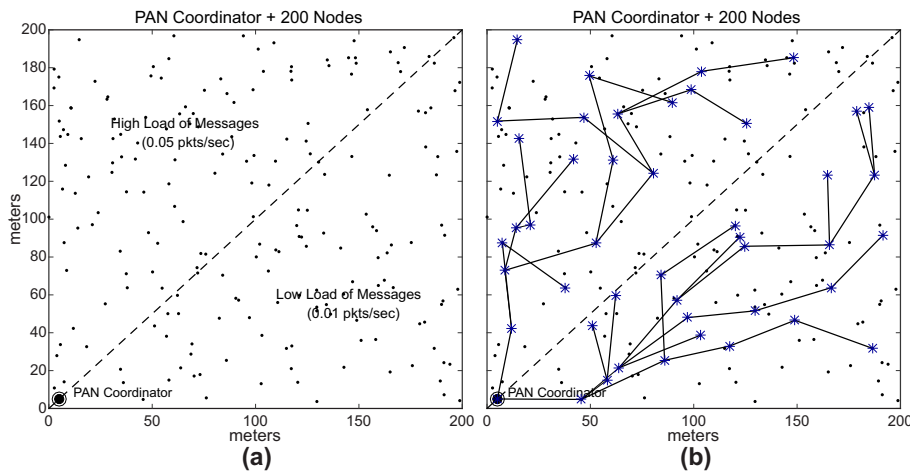


Figure 4.14: Configuration of the *conditioned Scenario*: (a) Distributed data rates in the two load zones; (b) Conditioned cluster-tree network.

Table 4.3: Summarising the different Simulation Scenarios.

Features	<i>Unconditioned Scenario</i>	<i>Conditioned Scenario</i>
Maximum number of child nodes (per cluster)	6 nodes	6 nodes
Maximum number of CCH for PAN coordinator	2 nodes	2 nodes
Maximum number of CCH (other CHs)	2 nodes	3 nodes
Set of data rates	0.05 and 0.01 pkts/s	0.05 and 0.01 pkts/s
Data rate distribution in the environment	randomly distributed	two well-defined zones

active period scheduling, we used a typical time division scheme. For the sake of simplification, for these experiments, we have used a bottom-up scheduling scheme, which prioritises the monitoring traffic (from leaf cluster-heads toward the PAN coordinator). Basically, the main difference between the bottom-up and top-down scheduling schemes is the protocol constraint, where the top-down scheme has more demanding beacon interval constraints than the bottom-up scheduling.

Regarding to some node's features, we have adopted the CC2420² radio model, which is compliant with the IEEE 802.15.4 standard. Furthermore, we adopted a linear energy model provided by Castalia and the initial energy for all nodes was set to 18.720 Joules (typical energy for two AA batteries). We also adopted the *unit disc* model as radio propagation model, where the range of the disk was defined to 55 m. For the interference model, we use a simple interference model provided by Castalia, where concurrent transmissions generate collisions at the receiver. Table 4.4 summarises the most important configuration and the main CSMA-CA parameters used in the simulations.

4.9.1.1 Performance Metrics

Basically, the aim of this simulation assessment is to apply different superframe allocation schemes and to evaluate the network behaviour according to well-defined metrics, both for

²Texas Instruments / Chipcon CC2420 Datasheet: <http://www.ti.com/product/CC2420/technicaldocuments>.

Table 4.4: Simulation Configuration.

Definition	Standard Value
Radio model	Chipcon CC2420
Initial energy (per node)	18.720 Joules
Simulation time (each experiment)	110.000 seconds
<i>aBaseSlotDuration</i>	60
<i>aNumSuperframeSlots</i>	16
<i>aUnitBackoffPeriod</i>	20
<i>macMinBE</i>	3
<i>macMaxBE</i>	5
<i>macMaxCSMABackoffs</i>	4
<i>macMaxFrameRetries</i>	3

the unconditioned and conditioned scenarios. For the sake of convenience, we used the following acronyms in the results, as follows:

- *Load-SDA: Proportional Superframe Duration Allocation* scheme based on the message load (Load-SDA). For this assessment, based on a series of performed simulation experiments, we have defined X value (Equation 4.29) to be 2 (two) messages, considering the main CSMA-CA parameters defined to their standard values (SD_{min} corresponds to 15,36 *milliseconds*, assuming a network with bit rate of 250 kbps, frequency band of 2.4 GHz, and one symbol as 4 bits).
- *Nodes-SDA: Proportional Superframe Duration Allocation* scheme based on the number of nodes (Nodes-SDA). In this case, similar rules as defined for Load-SDA scheme were considered.
- *STD-SDA: Standard Superframe Duration Allocation* scheme. For this scheme, all clusters have the same Superframe Duration (SD) and Beacon Interval (BI). This scheme is completely static and does not consider any proportional mechanism or message priority, assigning the same BI and SD for all cluster-tree network. The BI value was defined following the same constraint imposed to the proposed allocation schemes, according to Equation 4.11. The SD value was defined as the upper average of the SOs (SuperframeOrder), as defined by the *Load-SDA* scheme for all clusters (respecting the constraint defined by Equation 4.5).

In order to analyse the network behaviour for the three different schemes, the following performance metrics were used:

- *Message Discard Rate*: percentage of discarded messages due to buffer overflows in the cluster-heads, considering the number of discarded messages vs. the number of messages that arrived to the cluster-head.
- *Message Loss Rate*: percentage of lost messages during the communication, considering the number of messages successfully received at the destination node and the number of messages generated by the source node (encompasses both discarded messages and the messages lost due to collisions).

- *Average End-to-end Delay*: time interval between the data frame generation at the application layer of the source node and its reception at the application layer of sink node.
- *Energy Consumption*: average energy consumption of the overall network.

For each of defined allocation schemes, 10 different simulations were performed for each scenario (unconditioned and conditioned) with different sets of random variables. Therefore, the results presented along of the next subsection correspond to the average results obtained from this set of simulations.

4.9.2 Results and Discussion

Firstly, we present some information about the cluster-tree network formation for each of the defined scenarios. Table 4.5 shows the average number of generated clusters during the cluster-tree formation, the average maximum depth of the cluster-tree network, and the average number of children per cluster.

Table 4.5: Information about the network formation.

Information	<i>unconditioned Scenario</i>	<i>conditioned Scenario</i>
Average number of clusters	45	47
Average maximum depth	9	8
Average number of children per cluster	4	4

The main target of the proposed proportional SDA schemes is to allocate adequate communication resources (superframe durations and buffer sizes) for the cluster-heads, avoiding problems as congestion and message discards due to buffer overflows. Therefore, the buffer behaviour is an important performance metric to evaluate the behaviour of the proposed allocation schemes. Table 4.6 shows the average buffer size (number of messages) of the cluster-heads for each depth of the cluster-tree.

Table 4.6: Buffer size for cluster-heads (per depth).

Depths	unconditioned Scenario			conditioned Scenario		
	Load-SDA	Nodes-SDA	STA-SDA	Load-SDA	Nodes-SDA	STA-SDA
Depth 1	98	98	200	98	98	200
Depth 2	47	47	200	39	39	200
Depth 3	24	24	200	19	19	200
Depth 4	20	20	200	15	15	200
Depth 5	17	17	200	12	12	200
Depth 6	12	12	200	9	9	200
Depth 7	7	7	200	6	6	200
Depth 8	5	5	200	6	6	200
Depth 9	5	5	200	7	7	200
Depth 10	2	2	200	4	4	200

Note that the proposed allocation schemes define buffer sizes for the cluster-heads that are proportional to the supported traffic (or the supported number of nodes), considering the defined buffer constraint (Equation 4.13). As STD-SDA scheme does not provide any mechanism to adequately define the buffer sizes, we have set the length of the CH internal buffers to the total number of sensing nodes.

Within this context, Figure 4.15 shows the average rate of discarded messages due to buffer overflows for the overall network, when applying the proposed allocation schemes for the defined scenarios.

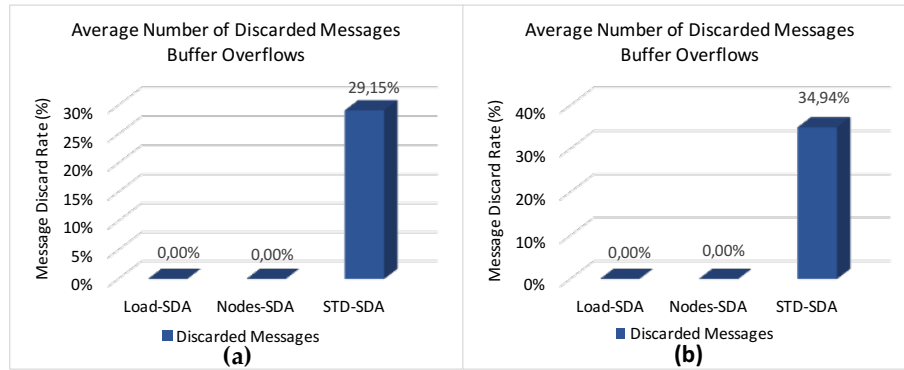


Figure 4.15: Average discard message rate due to buffer overflows: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

It can be observed in Figure 4.15 that both proposed allocation schemes adequately work for both communication scenarios, considering the defined set of message streams. All CHs were able to forward their messages and no message was discarded due to buffer overflows. On the other hand, the STD-SDA scheme discarded 30-35% of the messages due to the allocation of inappropriate superframe durations and also due to the inability of the cluster-heads to temporarily store the accumulated messages in their internal buffers, despite the larger number of allocated buffer resources.

Considering that the STD-SDA scheme discards multiple messages, Figure 4.16 shows the average number of discarded messages as function of the CHs' depth for this scheme. Note that, the number of discarded messages is higher for cluster-heads located in depth 1, followed of cluster-heads of depth 2, and so on. In fact, as all monitoring traffic is forwarded through the cluster-tree towards the PAN coordinator, the trend is that cluster-heads near the PAN coordinator will be more congested. This behaviour is observed for both the unconditioned and conditioned scenarios.

Figure 4.17 illustrates the total message loss rate (considering both collisions, and discards due to buffer overflows) for each of the defined allocation schemes. As it can be observed, the number of discarded messages due to buffer overflows strongly influences the number of lost messages, decreasing the number of successfully delivered messages. Comparing the results of both Figure 4.15 and 4.17, it is clear that, due to message collisions for the considered set of message streams, the message loss rate is around 21-22% for the three allocation schemes. The main reason for this high number of message losses is due to the (default) CSMA/CA parameters used for the simulation assessment. As previously shown [28, 111], the default parameter values used for *macMinBE*, *macMaxBE* and *CW* (Contention Window) can easily lead to a high number of message collisions, for a number of sensor devices as low as 6 devices per cluster-head.

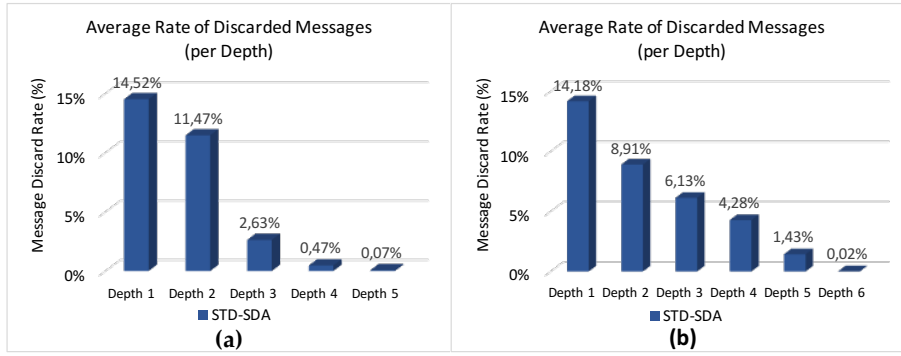


Figure 4.16: Average message discard rate (per depth) due to buffer overflows for the STD-SDA scheme: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

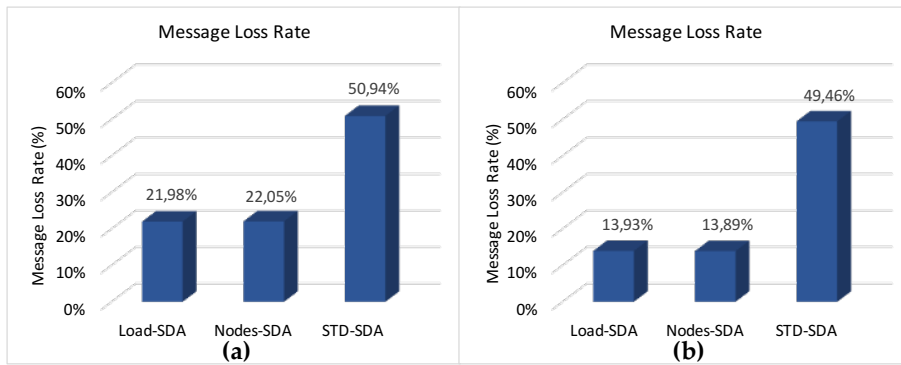


Figure 4.17: Message loss rate for the SDA schemes: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

With this set of simulations, we showed the importance of defining adequate active communication periods and buffer sizes for the cluster-heads. It can also be observed that the proposed allocation schemes have similar behaviour, showing that they significantly improve the network behaviour when compared to the standard allocation scheme.

Furthermore, we have also assessed the average end-to-end delay for the monitoring traffic, in order to evaluate the influence of the buffer size and superframe allocation schemes. Considering that the setting of beacon interval can directly influence the behaviour of the network (refer to Subsection 4.5.1), we have assessed the different possibilities for the BI adjustment and its impact upon the network behaviour (end-to-end message delays and energy consumption of the nodes).

Firstly, we have considered the case where the beacon interval is set to a value close to the shorter message period (longer beacon interval). Figure 4.18 shows the beacon interval and average sum of superframe durations defined for each of the allocation schemes.

Note that, as previously discussed in Section 4.8, the Load-SDA scheme generates a shorter sum of superframe durations than the Nodes-SDA scheme, by allocating well-adjusted superframe durations for cluster-heads. In turn, as the STD-SDA scheme allocates superframe duration based on the average of the SO parameter, the trend is that this scheme has a shorter sum of superframe durations, when compared with Load-SDA scheme (due the superframe duration being specified as a power of 2^{SO}).

Within this context, Figure 4.19 shows the average end-to-end delays for a network with

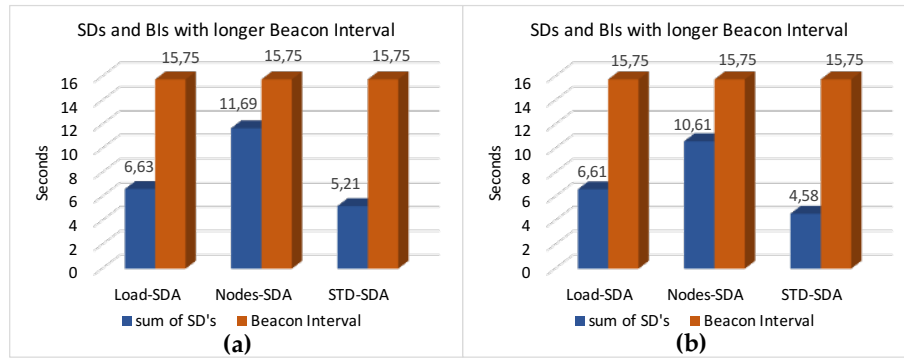


Figure 4.18: Beacon Interval and average sum of superframe durations for a network with longer beacon interval in the: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

longer Beacon Interval in both the unconditioned and conditioned scenarios. It can be observed that, the end-to-end delay for the STD-SDA scheme is remarkably higher than for both the Load-SDA and Nodes-SDA schemes. The main reason is that, for the STD-SDA scheme, messages are facing higher delays because they remain more time in the internal buffers of cluster-heads due to the network congestion. Therefore, messages need to wait more beacon intervals to be forwarded, increasing their end-to-end delays. In turn, as the proposed proportional allocation schemes define adequate communication periods and buffer sizes for the cluster-heads, the messages can flow along the routed tree until the sink node, and their end-to-end delays are smaller than the beacon interval, respecting the timing constraint. It can also be observed that the Load-SDA and Nodes-SDA schemes have a similar behaviour, showing that the active periods and buffer sizes are adequately assigned.

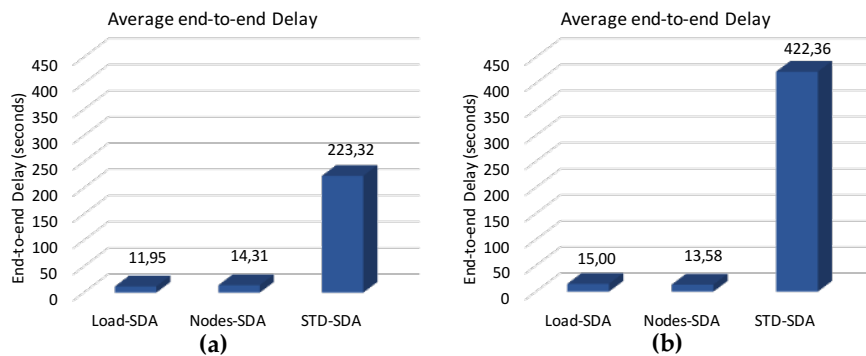


Figure 4.19: Average end-to-end delay for a network with longer Beacon Interval in the: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

Although presenting similar results regarding to end-to-end delays, loss rates, and buffer overflows, the Load-SDA and Nodes-SDA schemes present different results in what concerns energy consumption. The energy consumption is mainly related to two factors: the time interval during which the nodes remain active, and the activities performed by them. This way, as the Nodes-SDA scheme assigns higher superframe durations than the Load-SDA scheme, active periods of the nodes are larger and, consequently, the nodes spend more energy. Figure 4.20 illustrates the average total energy consumption for a network with longer Beacon Interval, considering both the unconditioned and conditioned schemes.

Note that, for both unconditioned and conditioned scenarios, the average energy

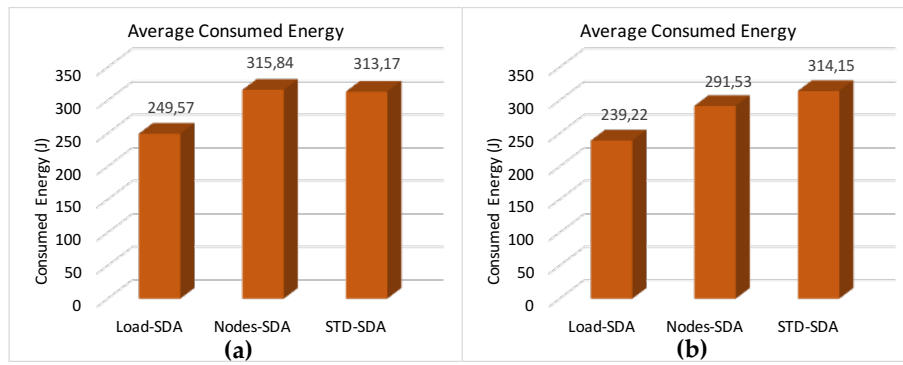


Figure 4.20: Average consumed energy for the SDA schemes: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

consumption for the Nodes-SDA scheme is larger than for the Load-SDA scheme. It is also important to highlight that the energy consumption of the STD-SDA scheme is proportional to the number of non-discarded message, and therefore there is a reduction of the energy consumption due to the large number of messages that is being discarded by buffer overflows.

We have also performed the same set of simulations, considering a network with shorter beacon interval. Figure 4.21 shows beacon interval and the average sum of superframe durations defined for each of the allocation schemes, considering a shorter beacon interval.

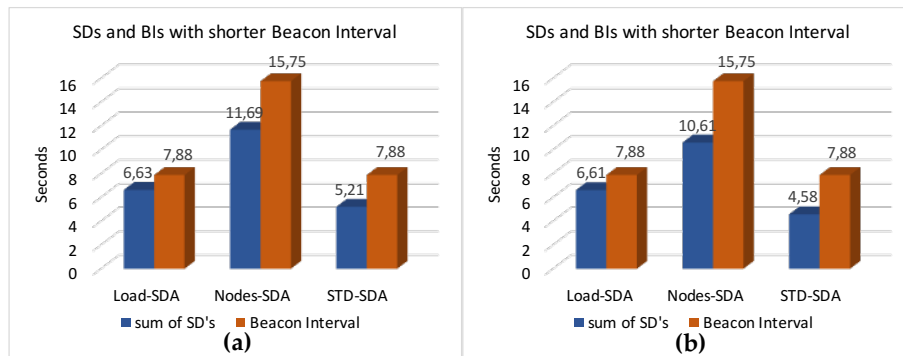


Figure 4.21: Average sum of superframe durations and beacon interval for the allocation schemes with adjustment of BI: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

In this case, the adjustment of the beacon interval was performed just for the Load-SDA and STD-SDA allocation schemes. For the Nodes-SDA scheme, if the beacon interval (in terms of beacon order) is decreased, the protocol constraint is no longer respected. Figure 4.22 shows the average end-to-end delay for a network with a shorter BI, for both the unconditioned and conditioned scenarios.

As it can be observed, the end-to-end delay for both Load-SDA and STD-SDA schemes is smaller when considering the reduction of the beacon interval, based on the considerations presented in Subsection 4.5.1. In fact, a shorter beacon interval corresponds to shorter end-to-end delays. On the other hand, the energy consumption will increase, due to the higher activation rate of the sensor nodes. Figure 4.23 illustrates the average total energy consumption for the SDA schemes, considering both unconditioned and conditioned allocation schemes and a shorter beacon interval.

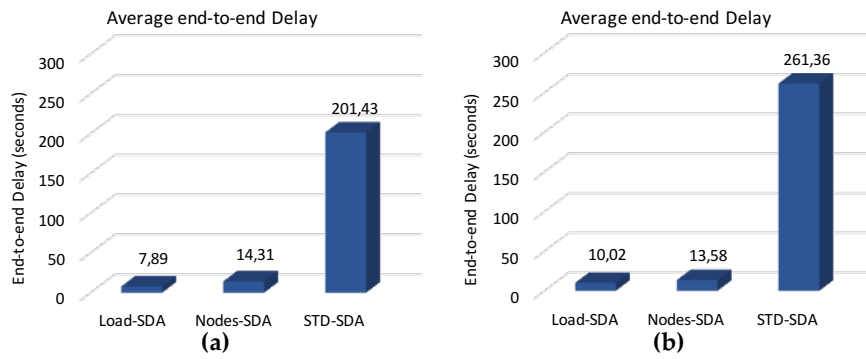


Figure 4.22: Average end-to-end delay for the monitoring traffic with a shorter beacon interval: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

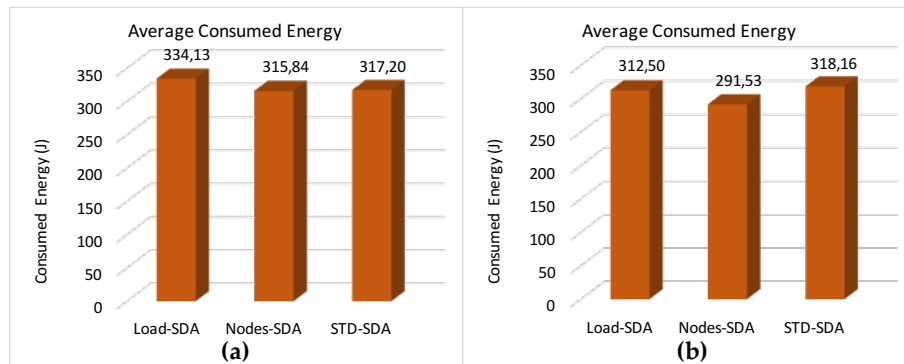


Figure 4.23: Average consumed energy for the SDA schemes with a shorter beacon interval: (a) *Unconditioned Scenario*; (b) *Conditioned Scenario*.

Note that, in spite of the higher activation rate of the sensor nodes, the average energy consumption for the Nodes-SDA and Load-SDA schemes is similar to the case with longer beacon interval (Figure 4.20). This behaviour highlights a problem that strongly influences the energy consumption. That is, the negative effect that a higher number of retransmissions due to the inadequate superframe duration allocation has upon the energy consumption of the overall network. Finally, it can be concluded that, for the case of shorter beacon intervals, the Load-SDA scheme has a better performance when compared to the Nodes-SDA and STD-SDA schemes (similar energy consumption, but shorter end-to-end delays).

Finally, in order to illustrate the difference between the Load-SDA and Nodes-SDA allocation schemes, Figure 4.24 illustrates the superframe duration configuration (in terms of superframe order, respecting the Equation 4.5) provided by the Load-SDA scheme, while Figure 4.25 illustrates the superframe duration configuration provided by the Nodes-SDA scheme for the same cluster-tree network, both for the conditioned scenario.

As it can be observed in Figure 4.24, the Load-SDA scheme allocates higher values of Superframe Order for the cluster-heads located in the high load zone, and lower values for the cluster-heads located in the low load zone. On the other hand, as it can be observed in Figure 4.25, the Nodes-SDA scheme allocates proportional values of Superframe Order for the cluster-heads of same depth, regardless of being located in a low load or a high load zone. Note that, as the Nodes-SDA scheme does not consider the load imposed by descendant nodes (it considers only the number of nodes), this allocation scheme may over-allocate higher

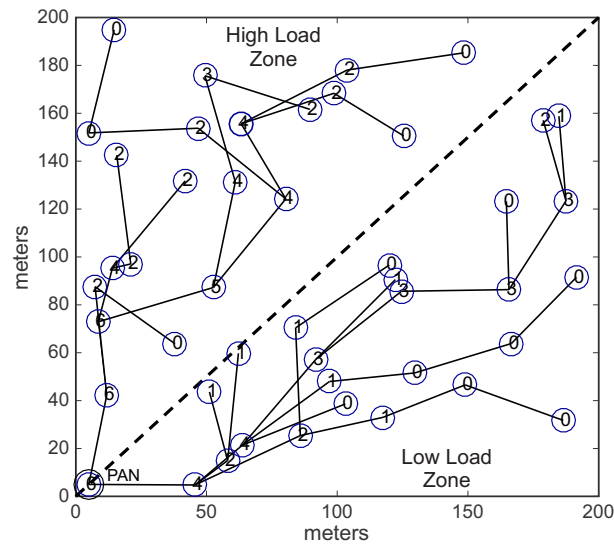


Figure 4.24: An example of Superframe Order (SO) configuration for all cluster-heads, using Load-SDA scheme (*conditioned Scenario*).

superframe durations for cluster-heads. Importantly, this behaviour was observed for all the simulation scenarios.

Table 4.7 presents the average superframe order values (per depth) defined by the Load-SDA and Nodes-SDA allocation schemes for both the high load and the low load zones of the conditioned cluster-tree.

Table 4.7: Average Superframe Order values (per depth) for both the high load and low load zones (*conditioned Scenario*).

Depths	Load-SDA scheme		Nodes-SDA scheme	
	High Load Branch	Low Load Branch	High Load Branch	Low Load Branch
Depth 1	6	4	6	6
Depth 2	4	2	4	4
Depth 3	3	1	3	3
Depth 4	3	1	3	2
Depth 5	3	1	3	2
Depth 6	2	1	2	2
Depth 7	2	1	2	1
Depth 8	1	1	1	1
Depth 9	1	-	1	-
Depth 10	1	-	1	-

In general, the Nodes-SDA scheme allocates similar superframe durations (in terms of the superframe order parameter) for cluster-heads of same depth for both zones, showing that the difference of message loads does not interfere in its allocation mechanism. In turn, the Load-SDA scheme allocates highest superframe duration values for cluster-heads located in the high load zone, while that cluster-heads located in the low load zone receive lower superframe duration values.

Finally, we have conducted a set of experiments to evaluate the scalability of the proposed

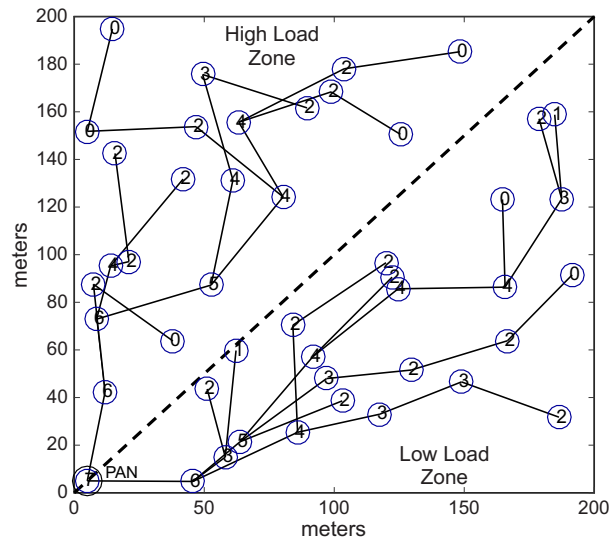


Figure 4.25: An example of Superframe Order (SO) configuration for all cluster-heads, using Nodes-SDA scheme (*conditioned Scenario*).

proportional SDA schemes, regarding the protocol constraint. For this, we used the *unconditioned Scenario* and increased the number of sensing nodes from 220 up to 380 nodes (with a step of 20 nodes). For each scenario, we have performed 10 simulations. Figure 4.26 shows the number of schedulable simulations for each of the proportional schemes, considering the different number of nodes.

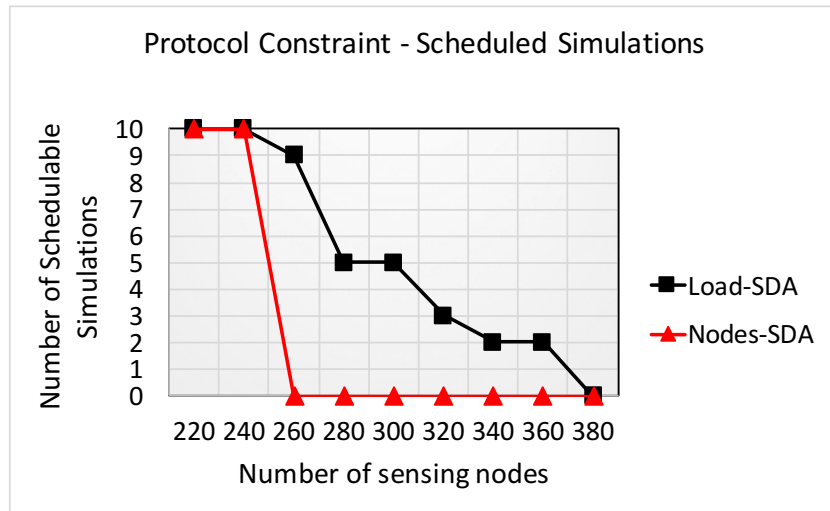


Figure 4.26: Number of schedulable simulations for the Load-SDA and Nodes-SDA schemes, considering different number of sensing nodes.

As it can be observed, for the used data rate configuration, the effectiveness of the Load-SDA scheme is higher than the Nodes-SDA scheme, showing that the Load-SDA scheme has a better scalability than the Nodes-SDA scheme.

From this simulation assessment, it can be concluded that the proposed proportional SDA schemes can adequately allocate the required communication resources for the cluster-heads (active communication periods and buffer sizes), avoiding traditional problems that occur in

cluster-tree networks, such as: network congestion, high end-to-end communication delays and discarded messages due to buffer overflows. The Load-SDA scheme presents better performance for cluster-tree networks than the other schemes, but it requires the knowledge of both the network topology and the data traffic loads. For the case where the data traffic load is not known, the Nodes-SDA scheme can be applied, obtaining excellent results regarding the end-to-end communication delays, number of discarded messages and energy consumption, when compared with static superframe duration allocation schemes.

4.10 Conclusion

The IEEE 802.15.4/ZigBee cluster-tree topology is one of the most suitable topologies to build wide-scale wireless sensor networks. However, these standards do not define mechanisms to adequately allocate communication resources to the cluster-heads (active communication periods and buffer sizes), in order to avoid common problems such as: network congestion near the PAN coordinator, discarded messages due to buffer overflows and high end-to-end communication delays. In this paper, we present a set of boundary equations for IEEE 802.15.4/ZigBee cluster-tree networks (protocol, buffer and timing constraints), which provide a set of guidelines for the proper allocation of communication resources. Within this context, we propose the use of two different proportional *Superframe Duration Allocation* (SDA) schemes: *Load-SDA* and *Nodes-SDA* schemes. The main target of these allocation schemes is to define adequate active communication periods and buffer sizes for the cluster-heads of the cluster-tree network. The Load-SDA scheme considers the message load imposed by descendant nodes to allocate superframe durations for cluster-heads. In turn, the Nodes-SDA scheme considers only the number of descendant nodes of the cluster-heads.

Simulation results show that the use of adequate superframe duration allocations and buffer sizes can improve several communication metrics, such as reducing both the number of discarded messages due to buffer overflows and the end-to-end message communication delays. Thus, the proposed schemes can be used by network designers to build efficient cluster-tree networks in what concerns the definition of network parameters and the configuration of message streams.

4.10.1 Future Considerations

As future considerations, we intend to add new mechanisms for the allocation schemes, such as: a) configuration of CSMA-CA parameters to improve the message throughput, and b) aggregation or information fusion mechanisms in order to decrease the number of messages to be transferred in congested areas of the network.

Acronyms

Table 4.8 shows the list of symbols, parameters and variables used along this work.

Table 4.8: List of symbols, parameters and variables.

Symbol	Description
α	<i>aBaseSuperframeDuration</i>
δ_i	Enough time to transmit the message M_i^k
η	Number of messages generated by the S_j message streams located in the descendant below nodes of CH_j
γ_i	Initial delay for message stream S_i
σ_i	Time interval immediately lower than the transmission time for one message M_i^k
Θ	Interference imposed by highest priority message streams
b	Maximum number of backoffs
CH_j	Number of hierarchically descendant nodes below
D	Radio data rate
i	Index of Message Streams
j	Index of Cluster-head
L_{ACK}	ACK length
L_{PACKET}	Frame data length
M_i^k	K-th message of Message Stream i generated by node N_n
n	Index of leaf nodes
N	number of hierarchically descendant nodes below the specific cluster-head
N_{nodes}	Number of sensor nodes
n_{CH}	Number of child cluster-heads
N_{CH}	Number of cluster-heads
n_l	Number of leaf child nodes
\mathbb{P}_c	Probability of a node to perform two consecutive CCA analysis
\mathbb{P}_{co}	Collision probability of two nodes
\mathbb{P}_s	Probability of a node to perform a CCA with the maximum backoff number
q	Probability that a node transmits a single data at any time of the CAP
r	Average backoff number
R_i	Worst-case response time for message stream S_i
\mathbb{S}	Set of Message Streams
\mathbb{S}_{below}^j	Set of message streams located in the descendant nodes of CH_j
S_i	Message Stream i
T_{ACK}	Acknowledgement transmission time
T_{BO}	Average backoff time
T_{BOL}	Backoff period length
T_{BOT}	Total backoff time
T_{PACKET}	Packet transmission time
$T_{radio_transition}$	Time duration that the radio takes to switch between the different operation modes
T_{TXD}	Total transmission time for a single data frame
U	Total Effective Utilisation
W	Local worst-case response time
X	Messages transferred during the minimum superframe duration
Y	Load imposed by all message streams hierarchically below the specific cluster-head

This page was intentionally left in blank

Alternative Path Communication in Wide-Scale Cluster-Tree Wireless Sensor Networks using Inactive Periods

This chapter presents a new alternative communication scheme, called Alternative-Route Definition (ARound), that is able to support peer-to-peer message-stream communication in cluster-tree WSNs. This chapter is a reproduction of the following manuscript:

LEÃO, Erico; MONTEZ, Carlos; MORAES, Ricardo; PORTUGAL, Paulo; VASQUES, Francisco. Alternative Path Communication in Wide-Scale Cluster-Tree Wireless Sensor Networks using Inactive Periods. **SUBMITTED TO:** *Sensors*, MDPI, 2016.

5.1 Abstract

The IEEE 802.15.4/ZigBee cluster-tree topology is being pointed out as a suitable topology to deploy wide-scale Wireless Sensor Networks (WSNs). However, one of its most relevant weaknesses is that all communication paths go through the PAN (Personal Area Network) coordinator, resulting in a potentially higher number of communication hops and higher network congestion. Within this context, this paper proposes the *Alternative-Route Definition* (ARound) communication scheme. The underlying idea of the ARound scheme is to setup alternative multihop communication paths between specific source and destination nodes, avoiding the congested cluster-tree paths. These alternative paths consider the shortest inter-cluster path between source and destination nodes, using a set of intermediate nodes to relay messages during their inactive periods in the cluster-tree network. Simulation results show that the ARound communication scheme can significantly decrease the end-to-end communication delay of messages and the energy consumption of nodes, when compared to the use of the standard cluster-tree communication scheme. Moreover, the ARound scheme is able to reduce the network congestion around the PAN coordinator, and therefore is also able to promote the reduction of the number of message drops due to queue overflow in the cluster-tree network.

5.2 Introduction

Wireless sensor networks (WSNs) are special *ad hoc* networks composed of a large number of low-cost, low-power, and low-rate wireless devices (i.e. sensor nodes) with capability for

sensing and, occasionally, actuating upon the environment where they are deployed. WSNs can be used to support the implementation of multiple types of monitoring applications, such as: health, structural and environmental monitoring, home automation, vehicular systems, military applications, and many others [6, 10]. In addition, due to their specific characteristics such as: flexibility, mobility, autonomous and collaborative operation, WSNs can be deployed in hazardous or hostile environments, where the human presence or wired systems are unsuitable.

Recent advances in Micro-Electro-Mechanical systems [10], microprocessors and low power radio technologies [6] enable the availability of wireless sensor devices increasingly robust, with more storage and processing capabilities, and at a lower price. In this context, the implementation of WSN wide-scale applications with energy-efficient and time-sensitive requirements have become an attractive research topic.

The IEEE 802.15.4/ZigBee standards [21, 68] define a set of networking topologies commonly used in WSNs. The IEEE 802.15.4 defines PHYsical (PHY) and Medium Access Control (MAC) sublayers for Low-Rate Wireless Personal Area Network (LR-WPAN), focusing on short-range operation, low-data rate, energy-efficiency and low-cost implementations. The IEEE 802.15.4 standard basically defines two types of devices: a Full-Function Device (FFD) and a Reduced-Function Device (RFD). The FFD is capable of serving as a Personal Area Network (PAN) coordinator, a coordinator, or a simple device, while the RFD is only capable of serving as a simple device. A Wireless Personal Area Network (WPAN) is composed of multiple FFD and RFD devices, with a unique FFD acting as PAN coordinator.

IEEE 802.15.4 defines two types of topologies: star and peer-to-peer. In the star topology, all sensor nodes directly communicate with the PAN coordinator. Thus, the network range is limited by the transmission range of its nodes, which is an impairment for wide-scale deployments. The peer-to-peer topologies allow the implementation of complex network formations, such as mesh and cluster-tree topologies.

The mesh networking topology uses a decentralized communication paradigm, allowing multihop routing from any node to any neighbour node. Mesh topologies allow network flexibility and provide lower complexity, high routing redundancy and good scalability [25, 26]. However, due to lack of synchronisation mechanisms, this type of topology does not allow nodes to enter into low power mode to save energy [27, 28], bounding the network lifetime. Moreover, it introduces additional complexity to ensure end-to-end connectivity [26]. These characteristics are undesirable for typical WSN-based monitoring applications, where power consumption and lower complexity are crucial issues.

The cluster-tree hierarchical topology enables setting-up wide-scale deployments through a mesh of multiple neighbouring clusters. Therefore, cluster-tree topologies are preferable for setting-up wide-scale monitoring applications.

The IEEE 802.15.4 standard does not describe by itself the required mechanisms to deal with cluster-tree topologies. It has been complemented by the ZigBee specification [21] that defines both network and application layers. In the network layer, ZigBee defines a set of rules enabling the network formation, addressing, and routing, providing mechanisms to build cluster-tree networks. These mechanisms employ a hierarchical addressing scheme and use tree-based routing algorithms. This type of topology has the benefits of using the beacon mode, which allows synchronisation, collision-free time slots, and duty-cycle operation. However, it still has some relevant limitations in what concerns real-time aspects and

energy-efficiency [42, 43], beacon frame collisions [26, 49, 52, 54, 56] and congestion control [118]. Additionally, as all cluster-tree paths go through the PAN coordinator, communications are prone to higher delays whenever the coordinator is not the final destination of the message streams. Thus, the design of new communication approaches is desirable, in order to guarantee the requirements imposed by typical wide-scale WSN applications.

5.2.1 Objectives and Contributions

In this paper, we propose the *Alternative-Route Definition* (ARound) communication scheme, that is able to deal with peer-to-peer communication in cluster-tree WSNs without relaying messages through the PAN coordinator (root node). The proposed scheme defines alternative communication paths able to support peer-to-peer message streams in a cluster-tree wireless sensor network. Such alternative paths are setup without interfering with the pre-defined cluster-tree communication. The underlying idea is to search for new inter-cluster mesh paths compatible with the previously defined beacon scheduling, using available border nodes of the clusters during their inactive periods. The target is to support source-to-destination message streams using these alternative paths, which may result in smaller end-to-end communication delays and smaller overall energy consumption, with the consequent increase of the network lifetime. We envisage the use of this communication scheme in real world applications such as precision agriculture and environmental monitoring applications, which are commonly deployed in wide-scale areas and handle a large number of message streams generated by a large number of nodes.

To the best of our knowledge, this is the first study that proposes the use of border nodes during their inactive periods to provide shortest communication paths, avoiding interferences with the previously scheduled cluster-tree communication. The main contributions of this paper can be summarized as follows:

- The definition of a new communication scheme for cluster-tree networks that avoids the use of default cluster-tree paths that go through congested areas of the network; instead, it defines alternative communication paths between source and destination nodes, allowing to save resources of the cluster-tree network. It can extend the cluster-tree network lifetime, because it saves energy from cluster-head nodes, which have a crucial role in keeping the network backbone working.
- The behaviour of the ARound scheme has been assessed and verified through simulation, highlighting its advantages in what concerns the reduction of both end-to-end communication delays and energy consumption of the overall network.

5.2.2 Organisation of this paper

This paper is organized as follows: Section 5.3 provides the required background for the proposed ARound scheme. Subsection 5.3.1 presents an overview of the IEEE 802.15.4/ZigBee cluster-tree topology and Subsection 5.3.2 presents the state-of-the-art on routing protocols for wireless sensor networks, highlighting applications for wide-scale

wireless sensor networks, cluster-tree topology and beacon frame scheduling. Section 5.4 presents the proposed ARound communication scheme, that enables the support of peer-to-peer message streams in IEEE 802.15.4/ZigBee cluster-tree networks, highlighting its main algorithms and protocol mechanisms. Section 5.5 presents a simulation assessment of the proposed communication scheme. Finally, in Section 5.6, some conclusions and future considerations are presented.

5.3 Background

5.3.1 IEEE 802.15.4/ZigBee Cluster-tree Topology

The cluster-tree topology is a special case of a peer-to-peer network. In a cluster-tree topology, nodes/devices are grouped in clusters, coordinated by a FFD node called Cluster-Head (CH). The CH provides synchronisation mechanisms for its associated nodes and centralizes all intra-cluster communication. The cluster-tree network formation is initiated by the PAN coordinator, which acts as coordinator for the network, being responsible for all network management activities.

According to the IEEE 802.15.4 standard, the simplest case of a cluster-tree network is a single cluster (coordinated by the PAN coordinator). New nodes may be allowed to create their own clusters, increasing the coverage of the network. Within this context, several neighbouring clusters can be used to build wide-scale cluster-tree networks, where the coordinators are connected by parent-child relationships, forming a multicluster hierarchical network structure. Figure 5.1 shows an example of an IEEE 802.15.4 cluster-tree network.

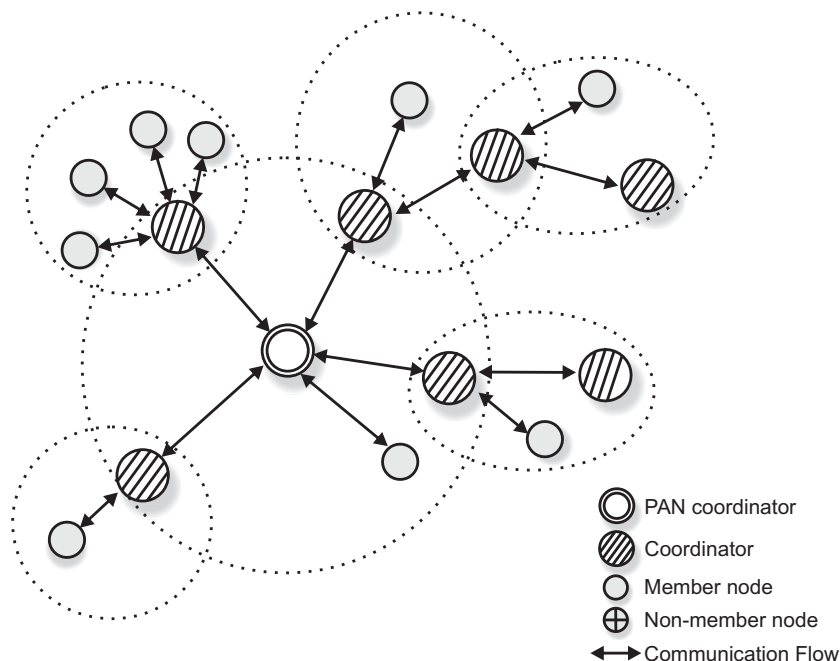


Figure 5.1: Cluster-tree networking topology.

Although considering cluster-tree networks, the IEEE 802.15.4 standard does not discuss network formation mechanisms. The ZigBee specification provides these mechanisms,

defining a hierarchical addressing scheme and the associated tree routing algorithm. In the ZigBee hierarchical addressing scheme, the network addresses are assigned based on address blocks. Thus, each cluster-head has its own block of sequential addresses, which are assigned to its child nodes. Based on this hierarchical addressing, ZigBee also provides a deterministic tree routing scheme. In this scheme, routing is based on the destination address. Case the destination address is a descendant node, the packet is forward to the corresponding child node; otherwise, the packet is forward to the parent node.

In cluster-tree topologies, the network operates in a beacon-enabled mode. In this mode, communication exchanges are organized according to a structure called *Superframe*. A superframe is bounded by beacon frames periodically transmitted by the coordinators (cluster-heads). These beacon frames synchronise the associated nodes, identify the PAN and describe the superframe structure. Figure 5.2 illustrates the superframe structure.

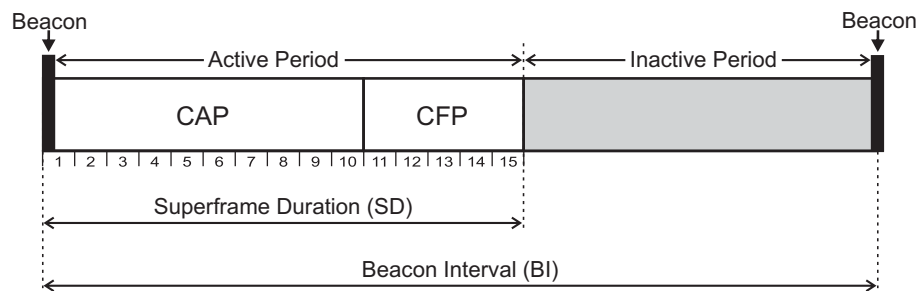


Figure 5.2: IEEE 802.15.4 Superframe structure

The superframe structure is described by the *macBeaconOrder* (BO) and *macSuperframeOrder* (SO) parameters, where *Beacon Interval* (BI) and the *Superframe Duration* (SD) are defined as follows:

$$BI = aBaseSuperframeDuration \times 2^{BO}(\text{symbols}), \quad (5.1)$$

$$SD = aBaseSuperframeDuration \times 2^{SO}(\text{symbols}), \quad (5.2)$$

where $0 \leq SO \leq BO \leq 14$.

BI defines the interval at which the coordinator periodically transmits beacon frames. In turn, the SD parameter defines the length of the active portion of the superframe. The *aBaseSuperframeDuration* parameter defines the minimum duration of the superframe when SO is 0. The IEEE 802.15.4 standard defines this parameter with 960 symbols duration, which corresponds to 15.36 ms (assuming a network with bit rate of 250 kbps, frequency band of 2.4 GHz, and one symbol as 4 bits).

Each superframe is divided in two parts: the active and inactive periods. During the inactive period, the coordinator and associated nodes can enter in low power mode to save energy (sleep mode). During the active period, nodes can communicate with the coordinator. The active part (communication period) comprises two periods: *Contention Access Period* (CAP) and *Contention-Free Period* (CFP).

In the CAP period, if a device wishes to communicate, it will contend with other devices using a slotted *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) procedure

to access the channel. The CFP is defined for applications that require low latency or specific bandwidth. In the CFP period, the coordinator allocates *Guaranteed Time Slots* (GTS) for specific nodes. In these slots, nodes can transmit data without contending for the channel access. The coordinator can allocate up to seven GTSs, and a GTS is allowed to occupy more than one slot period [68].

In the cluster-tree topology, a coordinator node (excluded the PAN coordinator) must keep a synchronisation between its own active period (acting as coordinator node) and the parent's active period (acting as child node). Also, for each cluster, low duty cycles can be activated to save energy, setting a value of SO smaller than BO.

In cluster-tree networks, sending beacon frames without any special care on timing issues may result in collisions among beacons from neighbour clusters [26]. Thus, it is necessary to implement inter-cluster synchronisation mechanisms to avoid this problem. Two types of beacons collisions are possible: direct or indirect. In a direct beacon collision, two or more coordinators (in the transmission range of each other) transmit their beacon frames at the same time. In the indirect beacon frame collision, two or more coordinators are hidden-nodes from each other and send messages to an overlapped node [26].

This paper is focused on cluster-tree networks defined by the ZigBee specification [21], due to some specific features, such as: suitability to deploy wide-scale networks, energy-efficiency and time-sensitive message guarantees.

5.3.2 Related Work

In recent years, due to the high demand of applications based on wireless sensor networks, there have been a large number of works addressing cluster-tree WSNs. Multiple issues are being addressed, such as: network formation schemes, routing protocols, real-time requirements, reliability and availability issues, energy-efficiency, scalability, congestion control and beacon frame scheduling. Each of these issues have their own challenges and special considerations. Additionally, with the increased demand for wide-scale applications that need to operate with time-sensitive data and must be energy-efficient, the design of wide-scale WSNs is becoming more challenging due to the large number of constraints that need to be simultaneously satisfied.

The state-of-the-art indicates that the utilisation of hierarchical clustering protocols has some advantages over flat protocols for wide-scale applications with respect to scalability [48, 119, 120], energy-efficiency [26, 52, 53, 119, 120] and time-sensitive data operation [26, 49, 52, 53, 120, 121]. Therefore, several works have been published in the literature, encompassing clustering for WSNs, with different concerns. As an example, there are several surveys that summarize popular clustering protocols and analyse the strengths and weaknesses of available routing protocols based on different metrics [14, 30, 60, 61, 122, 123]. For example, the LEACH (*Low Energy Adaptive Clustering Hierarchy*) protocol [84] is one of the most cited hierarchical protocol in the literature, and has motivated the design of several other protocols. In LEACH, sensor nodes are grouped in clusters for sending data to their cluster-heads, which aggregate data and send it to the base station. In order to balance the energy consumption of the sensor nodes, this protocol randomly selects a set of different cluster-heads during each round. Also, Kumar et al. [119] proposed an Energy Efficient Heterogeneous Clustered Scheme (EEHC) based on the

LEACH protocol, considering the heterogeneity of network resources such as node energy. However, since it uses single-hop routing, the cluster-heads send directly data to the base station, which leads to an increase of energy consumption. Thus, this type of communication can be infeasible for wide-scale environments.

As the communication scheme proposed in this paper addresses setting-up new communication paths in wide-scale cluster-tree networks, in order to support time-sensitive message streams, this state-of-the-art is focused in relevant approaches regarding cluster-tree formation, both its synchronisation and its congestion control and data communication support in wide-scale cluster-tree networks.

Within this context, Zhu et al. [124] proposed a *tree-cluster-based data-gathering algorithm* (TCBDGA) using a mobile sink to improve the non-uniform energy consumption of WSNs. The basic idea of this algorithm is to build a cluster-tree network considering characteristics of the nodes, such as their residual energy, distance to base station and number of neighbours. After this, the network is decomposed in several sub-trees and each sub-tree has its own collect node. The mobile sink is responsible to collect data from nodes. This algorithm is a location-aware based algorithm, which implies more energy consumption and complexity.

Choi and Lee [43] implemented a multihop GTS mechanism for IEEE 802.15.4 beacon-enabled networks. The aim of this work is to allocate GTS slots from the requesting nodes to the sink node along the cluster-to-cluster path. In turn, Felske et al. [99] proposed the GLHOVE approach for cluster-tree networks to maximize the number of received messages in the base station, prioritizing specific clusters of the network. GLHOVE comprises distributed algorithms to guarantee fairness (balanced message delivery per cluster) and, at the same time, to avoid congestion and improve the network lifetime. However, none of these approaches consider any message priority mechanism.

Khatiri et al. [125] and Kim et al. [46] proposed approaches to decrease the number of hops to reach the sink node using the neighbouring table defined in the ZigBee standard. The purpose is to reduce the routing costs, improving the energy consumption, network lifetime and end-to-end delays. In [125], each node keeps a neighbouring table with relevant information about the neighbouring nodes, such as: depth, link quality and device type. Thus, the algorithm defines the shortest path using three criteria: minimum hop count, minimum congestion and maximum link quality. Each criterion has its weight and the next hop (neighbour node) is selected based on a minimum cost function. In [46], the algorithm inspects the neighbouring table in order to find a node with a shortest tree path until the destination node and selects it as next hop. Although these algorithms define shortest paths until the sink node, they do not consider any information about network scheduling and how the data communication is performed.

Misic [58] presents a new approach that uses border nodes (slave-slave bridge) in order to interconnect neighbouring clusters. The slave-slave bridge nodes listen to the beacons generated by source and sink clusters. During the source cluster active period, the bridge node can receive data from the coordinator, and during the sink cluster active period, the bridge node delivers its data to the sink coordinator. Considering this approach, the cluster-tree path is not used. Although using bridge nodes to avoid the cluster-tree path, data is transmitted during the clusters' active period, and therefore, end-to-end delays are still dependent of the cluster scheduling.

Also, Huang et al. [126] proposed an *adoptive-parent-based* framework for ZigBee networks, to increase the bandwidth between the source and sink nodes. For this, a specific

cluster-head can request bandwidth from neighbour cluster-heads (adoptive parents) during a given time period. However, even using additional paths to transmit message between source and sink nodes, these additional paths are cluster-tree paths and therefore the problem of a higher energy consumption of the cluster-heads still remains.

Regarding beacon frame scheduling, some relevant works have also been presented in [26, 49, 52–54, 127]. The key idea of these works is to schedule the clusters' active period, considering the main constraints imposed by the message streams supported by the cluster-tree network.

Toscano and Lo Bello [49] and Abdeddaim et al. [127] follow a multichannel approach to avoid overlapping cluster collisions, while maintaining the cluster connectivity. In [49], the proposed approach schedules adjacent clusters in alternate timeslices, using different channels to avoid cross-channel interferences. Thus, while a coordinator schedules its superframe, its adjacent clusters can not schedule their owns. Therefore, collisions are avoided between neighbour clusters. Following this approach, a specific cluster is able to receive the parent's superframe in a given timeslice and schedule its superframe in an adjacent timeslice. The clusters are scheduled within a major cycle, which is cyclically repeated. In order to avoid inter-cluster collisions, a set composed of the PAN coordinator and all the clusters that can be reached in an even number of hops are scheduled in a given timeslice; then, all other clusters are scheduled in alternate timeslices. Abdeddaim et al. [127] proposed an IEEE 802.15.4-based cluster-tree formation protocol, named Multi-Channel Cluster Tree (MCCT), which uses multiplexed transmissions across different channels, in order to avoid beacon collisions. In addition, it uses a shared control channel for the cluster-tree construction and maintenance operations. Thus, a node only scans the control channel to request an association. The joining node defines its parent cluster-head based on the number of associated children. The cluster-head with lower number of children is selected. Thus, the association process occurs during the active period, using the channel of the specific cluster-head. These works use multi-channel approaches in order to avoid beacon collisions, which add overheads related to the channel's maintenance and control.

Koubaa et al. [26] proposed the Time Division Beacon Frame Scheduling (TDBS) approach, which defines a Superframe Duration Scheduling (SDS) algorithm for cluster-tree networks. In this approach, superframe duration and beacon frames of a given cluster are scheduled in the inactive periods of its neighbour clusters to avoid inter-cluster interferences. The schedulability condition of the SDS is both necessary and sufficient, and is obtained considering the duty cycle information of nodes [49].

The TDBS approach defines minor and major cycles to schedule a cluster set with different superframe durations and beacon intervals. All clusters are organized within the defined major cycle, based on the Least Common Multiple (LCM) of the beacon periodicities for all the clusters. The major cycle is divided in minor cycles, which are used to sequentially fit all the clusters. In this approach, clusters are organized in an increasing order of beacon periodicities. To break ties, these clusters are organized in a decreasing order of superframe durations. Clusters are organized sequentially within the minor cycles, until reaching the end of the major cycle. TDBS defines the start time for all the clusters in a collision-free scheduling scheme. However, it does not consider any message stream prioritisation and therefore may not be adequate to support time-sensitive applications.

Hanzalek and Jurcík [52] present a Time-Division Cluster Scheduling (TDCS) mechanism to avoid inter-cluster collisions and to meet all end-to-end deadlines of time-bounded

message streams. This mechanism employs a pure time-division scheduling approach, avoiding the inter-cluster collision problem. Besides, it aims to define the maximum TDCS period (major cycle) in order to minimize the energy consumption of the nodes. This is a challenging task because it is necessary to consider the message stream requirements, meeting all their end-to-end deadlines. To solve this problem, the authors formulate the TCDS approach as a cyclic extension of the Resource Constrained Project Scheduling with Temporal Constraints (RCPS/TC), which defines a feasible schedule considering the temporal and resource constraints of a set of tasks. After modelling this problem, they use an integer linear programming algorithm to solve the scheduling problem.

The main weakness of TDBS and TDCS approaches is the off-line scheduling performed during the setup of the network, assuming static network conditions. Thus, Severino et al. [53] presented an interesting approach to modify the scheduling at run-time, in order to provide QoS (Quality of Service) to message streams. Basically, they proposed the use of two different techniques: Dynamic Cluster Re-ordering (DCR) and Dynamic Bandwidth Re-allocation (DBR). The DCR technique is used to re-order at run-time the cluster scheduling. The clusters involved in the data streams are re-ordered, in order to minimize the traffic latency. This re-scheduling is performed based on the priority of the data streams, in order to decrease the end-to-end delays. In turn, the DBR technique is used to change the superframe duration of the clusters, increasing the bandwidth of the clusters involved in the message stream. This technique uses the free space (not used to allocate clusters) and distributes it among the involved clusters. If there is no available free space, it tries to reduce the bandwidth of the non-involved clusters, in order to re-distribute it among the involved clusters. Unfortunately, this communication approach only considers the use of cluster-tree paths.

Differently of the TDBS and TDCS approaches, Yeh and Pan [54] proposed an efficient beacon scheduling to support low-latency *upstream* and *downstream* traffic. The authors formulate the *Low-latency Two-way Beacon Scheduling* (LTBS) problem for ZigBee networks. In this problem, the nodes try to get slots for upstream and downstream traffic, while avoiding interferences from other clusters. LTBS modifies the IEEE 802.15.4 original superframe structure, in order to allow each cluster-head to broadcast two beacons: one for upstream direction and another for downstream direction. The authors propose two algorithms (centralized and distributed) to assign interference-free upstream and downstream slots to reduce the network latency. Basically, the difference between the centralized and distributed approaches is the sequencing order. The centralized algorithm assigns slots using a bottom-up approach. On the other hand, the distributed algorithm assigns slots using a top-down approach. This proposal minimizes the network latency without considering any message stream prioritisation.

Although improving the performance of cluster-tree networks, the proposed communication schemes do not address the definition of new communication paths and are limited to the cluster-tree routing and its inherent problems, such as: network congestion near the PAN coordinator, poor bandwidth allocation to the different clusters, and higher energy consumption of crucial nodes.

5.4 Alternative Paths for Message Streams in Cluster-Tree WSNs

In this paper, we propose a new communication scheme for IEEE 802.15.4/ZigBee-based cluster-tree wireless sensor networks, called *Alternative-Route Definition* (ARound). The underlying idea is to define alternative communication paths that avoid congested areas of the network to support peer-to-peer message streams. These alternative communication paths avoid interferences with the previously defined cluster-tree paths, by carefully selecting the timing instants when messages are transferred. The proposed scheme is well suited to support wide-scale applications requiring the support of some peer-to-peer message streams with high bandwidth requirements.

The use of ARound communication scheme can improve several performance metrics of cluster-tree networks. Firstly, by defining alternative communication paths, the ARound scheme may decrease network congestion around the PAN coordinator and also the energy consumption of the overall network. Secondly, by selecting shortest inter-cluster paths, ARound can potentially decrease end-to-end communication delays of message streams. Lastly, ARound may also guarantee a minimum level of Quality-of-Service (QoS) for peer-to-peer message streams, as the involved nodes will access the medium without contention.

In order to establish the new source-to-destination paths, the ARound communication scheme adds new functionalities for specific cluster-tree nodes and defines two main algorithms: *Alternative-Path Definition* (ARound-Def) and the *Alternative-Path Activation* (ARound-Act). In the next subsections, we describe both the network model and the ARound communication algorithms.

5.4.1 Network Model

Assume a set of sensor nodes randomly deployed in a wide-scale environment. These nodes are organized into clusters according to the IEEE 802.15.4/ZigBee cluster-tree topology [21, 68]. For the sake of simplicity, assume that this set of clusters is static and the network topology does not change along the network lifetime (there are no mobile nodes). Thus, the network is composed of N coordinator nodes (including the PAN coordinator), that act as Cluster-Heads (CHs) of their clusters and periodically send beacon frames to synchronise their child nodes, such that:

$$CH_i = (SD_i, BI_i), \quad 1 \leq i \leq N, \quad (5.3)$$

where i is a unique identifier for the specific cluster, SD_i corresponds to the *Superframe Duration* and BI_i is the *Beacon Interval* of the cluster-head CH_i . We assume that values for BO and SO are defined according to the load supported by each of the branches of the cluster-tree, in order to have a balanced service for the supported message streams. We also assume that the cluster's active periods were previously scheduled, considering a classical time division superframe scheduling approach, as the one proposed by [26]. Therefore, there are no beacon frame collisions caused by overlapping clusters. Importantly, for each node is assigned a hierarchical global address and all nodes operate over the same communication channel.

Figure 5.3 (real simulation) illustrates an example of this random cluster-tree network. The PAN coordinator is located in the centre of the wide-scale communication environment and it is denoted in the figure simply as PAN. The dot marks represent leaf nodes and the asterisk marks represent cluster-head nodes.

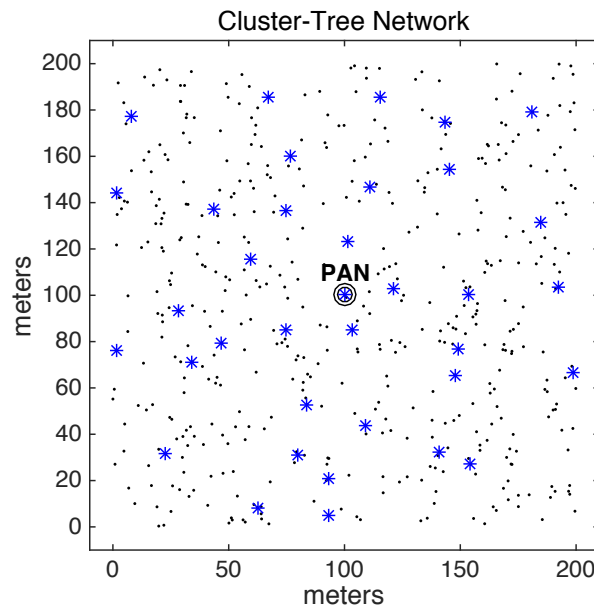


Figure 5.3: Cluster-tree network obtained from simulations.

Figure 5.4 shows a typical time division superframe scheduling scheme. As can be seen, the cluster's active periods are organized along the time, in order to avoid interferences. Thus, during the active period of their clusters, sensor nodes can communicate with their coordinators (cluster-heads) and vice-versa. During the inactive periods, sensor nodes can sleep to save energy.

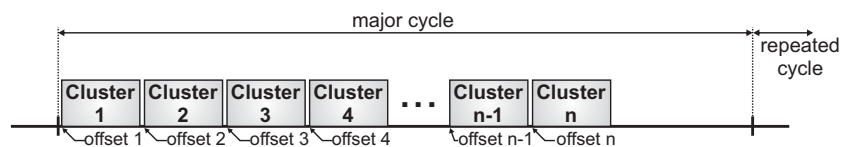


Figure 5.4: Cluster's active period scheduling along the time.

Figure 5.5 shows the duty cycle for the specific cluster 3, considering a cluster-tree time division superframe scheduling. Note that the active period of cluster 3 starts at instant *offset 3*, where all of its sensor nodes wake up to sense the environment and to communicate. During the remaining inactive period, these nodes can sleep to save energy.

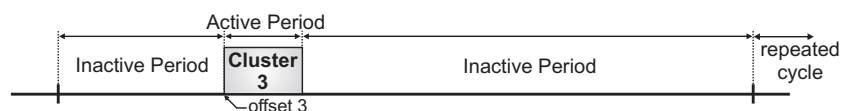


Figure 5.5: Duty cycle for the specific cluster 3, showing its active and inactive periods.

According to the cluster-tree communication algorithm, any message will be routed following the tree-based path, i.e., data packets will have paths defined by parent-child

relationships. Thus, all communications will go towards the PAN coordinator node, which results in higher energy waste, higher number of hops along the path, and higher delays [46, 99].

The proposed ARound communication scheme uses two algorithms (Alternative-Path Definition and Alternative-Path Activation algorithms) to establish an alternative path for a known message stream. Importantly, these two algorithms are performed by the PAN coordinator and use available resources during the inactive periods of the cluster-tree network. In the next subsections, we detail these two communication algorithms and the related ARound structures.

5.4.2 Alternative-Path Definition Algorithm

The *Alternative-Path Definition* (ARound-def) algorithm is performed by the PAN coordinator to find an alternative path for a given message stream. This way, after receiving a source-to-destination communication request, instead of considering the tree path, the PAN coordinator uses the available sensor nodes to define a *shortest inter-cluster path* between the source and destination nodes. This shortest inter-cluster path is related to the smaller number of overlapping clusters that the message stream should traverse from the source to the destination. Currently, the *number of clusters* is being used, but other metrics could be also used to define the shortest path, e.g. *link quality*, *bandwidth* or *energy capacity*.

Consider an example of a cluster-tree network, where the pair source-destination is located in specific positions in the communication environment as shown in Figure 5.6.

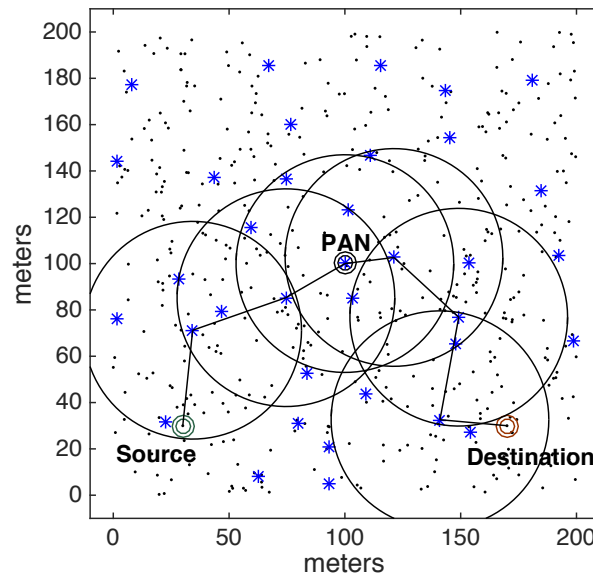


Figure 5.6: Standard cluster-tree path between the source and destination nodes.

Figure 5.6 illustrates the cluster-tree path between the source and destination nodes, where the message stream is required to go through the PAN coordinator along the tree path till the destination node. Along this path, each node must wait for the active period of their parent cluster, in order to communicate toward the cluster-head and send the data packets.

According to this forwarding strategy, a message stream may experience high end-to-end delays, which are highly dependent of the beacon scheduling strategy (favouring bottom-up or top-down traffic, but not both) and of the network congestion behaviour near the PAN coordinator.

Instead, the ARound communication scheme defines a different path, exploiting overlapping clusters, and building a shorter inter-cluster path between source and destination nodes. Figure 5.7 illustrates an ARound path for the same example. Note that the defined path also traverses some clusters but does not necessarily use the same cluster-head nodes to relay data packets from source to destination nodes.

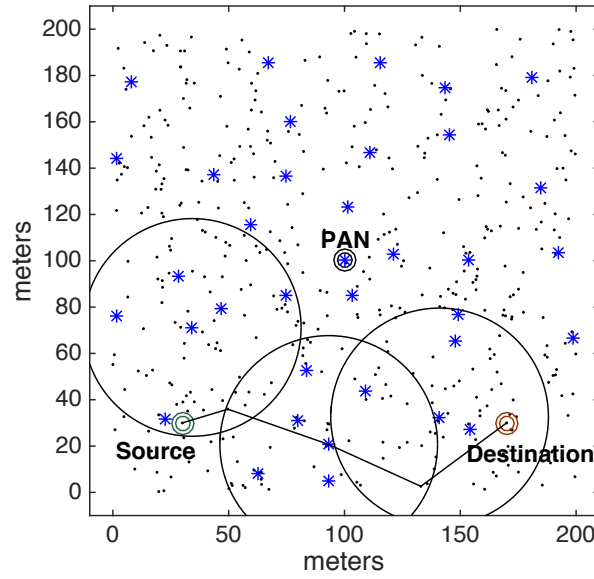


Figure 5.7: Alternative ARound path between the source and destination nodes.

The ARound communication scheme defines a set of special nodes: the *ARound Repeater Nodes* (RP). This type of nodes are responsible for relaying data frames among overlapping clusters along an ARound path, acting as border nodes between clusters. An RP node is a full function device that sits in the transmission range of two or more clusters, independently of its functionality in the cluster-tree.

In order to represent the repeater nodes and define the collision domain of the cluster-tree network, we use an adjacency list, which is a structure of linked lists used to represent finite graphs. Thus, considering a cluster-tree network with N cluster-heads, the ARound scheme defines an adjacency list $C = |V(i)|$, for $1 \leq i \leq N$, where $V(i)$ corresponds to a linked list of all neighbour clusters, being the cluster coordinated by CH_i . Based on the adjacency list C , we defines a list of edges $E = |RP_{i,j}|$, where $RP_{i,j}$ corresponds to a list of repeater nodes between clusters CH_i and CH_j (for $1 \leq j \leq N$). Importantly, both the adjacency list and the RP nodes are defined during the formation phase of the cluster-tree network through message exchanges.

The first step of the ARound-def algorithm (lines 2-5 in Algorithm 3) is to use adjacency lists to obtain an inter-cluster graph. Based on this graph, it has all possible inter-cluster paths from a source node to a destination node. With this, the PAN coordinator defines the *shortest*

inter-cluster path between the pair of source and destination nodes, according to a *shortest path algorithm* (e.g. the *Dijkstra's algorithm* [128]).

Algorithm 3: Alternative-Path Definition (ARound-def) Algorithm.

```

input : Adjacency List C;
        MStreamm: [SRCm, DESTm];
output: RPnodes = set of Repeater nodes;

1 // First Step: to define the shortest inter-cluster path
2 Clustersrc = get_Cluster(SRCm); // Get the cluster ID of the source node
3 Clusterdest = get_Cluster(DESTm); // Get the cluster ID of the
  destination node
4 ARound_Pathm = Dijkstra_Algorithm(C, Clustersrc, Clusterdest);
5 cluster2clusterm = set of pairs of ARound clusters for ARound_Pathm;

6 // Second Step: to select the repeater nodes
7 for (each pair [Clusteri, Clusterj] ∈ cluster2clusterm) do
8   if (leaf_RP(Clusteri, Clusterj) ≠ ∅) then
9     RPnodesi,j = selectRP(leaf_RP(Clusteri, Clusterj));
10  else
11    RPnodesi,j = selectRP(cluster-head_RP(Clusteri, Clusterj))
12 return RPnodes;

```

After defining the shortest inter-cluster path (ARound path), the next step is to select the repeater nodes that will relay data packets along this path. In order to structure and optimise this selection, we decompose this path (set of clusters) as a sequence of pairs of clusters (called *pairs of ARound clusters*, lines 4 and 5 in Algorithm 3). For each pair of clusters, the selected repeater nodes are responsible for relaying data frames from the initial cluster to the next cluster. Figure 5.8 illustrates the selection of repeater nodes for the pairs of ARound clusters. As it can be seen in Figure, the ARound-def algorithm selects a specific repeater node for the first pair of clusters. Following the ARound path, the algorithm selects repeater nodes for the second pair of clusters and, finally, the destination node is reached.

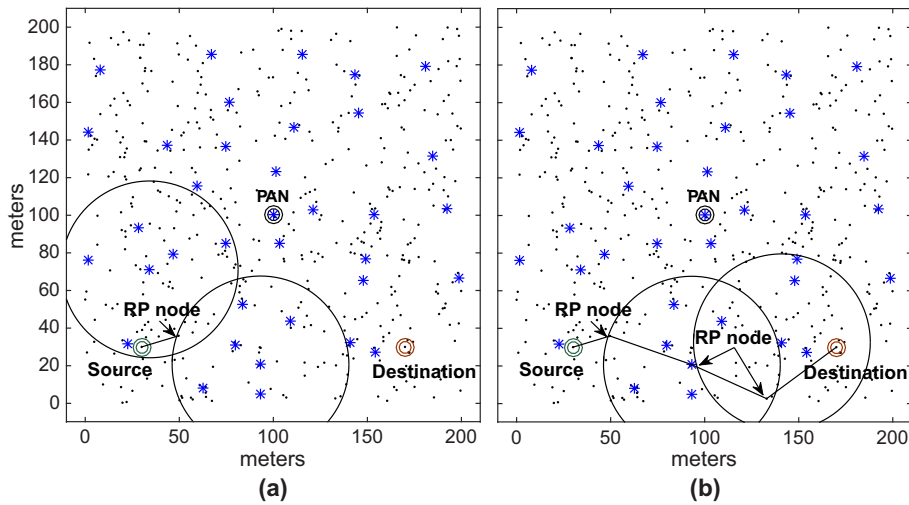


Figure 5.8: Applying the ARound-def Algorithm: (a) Selecting repeater nodes for the first pair of clusters; (b) Selecting repeater nodes for the second pair of clusters.

The selection process of repeater nodes uses an optimisation algorithm (lines 7-11 in Algorithm 3). AROUND keeps two lists of repeater nodes for each pair of clusters: list of *leaf* RPs, which contains RP nodes that act as leaf nodes in the cluster-tree network; and a list of *cluster-head* RPs, with RP nodes that act as cluster-heads. As long as we are dealing with full function devices, both type of RP nodes can act as repeater nodes during the AROUND communication. The AROUND scheme firstly inspects the list of *leaf* RPs, to save energy of cluster-heads that are responsible to keep the cluster-tree topology. When there are no available leaf RPs, the algorithm selects a cluster-head node to act as RP node in the AROUND path. The *selectRP* function (lines 9 and 11 in Algorithm 3) implements the mechanisms to select an RP node among the available RP nodes in each list. In this work, the *selectRP* function randomly selects a repeater node. However, other RP selection policies can be defined to implement this function, considering a specific selection criteria such as: remaining energy and physical position (location-aware protocols) [129, 130].

After selecting the repeater nodes for each pair of AROUND clusters, the next step performed by the PAN coordinator is to define their activation times using the *Alternative-Path Activation* algorithm.

5.4.3 Alternative-Path Activation Algorithm

The *Alternative-Path Activation* (AROUND-act) algorithm defines the activation instants for the selected RP nodes, and is also performed by the PAN coordinator. The activation instant of an RP node is a crucial issue. Due to the current cluster-tree communication, the AROUND communication must be adequately scheduled; otherwise, it can generate collisions and interferences with the cluster-tree communication, which would cause serious problems for the cluster-tree network, such as loss of synchronisation, loss of messages, or partitioning of the network. To prevent these problems, the AROUND communication scheme activates the RP nodes during their inactive periods and only when there is no activity in the neighbourhood.

We define the *AROUND Activity Period* as the period during which an RP node can be activated and perform AROUND communication without interfering with the cluster-tree communication. The definition of the *activity period* is based on two considerations: 1) inactive period of this specific node in the cluster-tree, and; 2) active periods of the one-collision clusters (neighbours) of this node (considering its collision domain). Within this context, we define two types of *AROUND activity periods*, as follows:

- **Global AROUND Activity Period (GAP):** period during which there are no active clusters in the cluster-tree network;
- **Local AROUND Activity Period (LAP):** period during which there are no active clusters in the neighbourhood of the RP node.

The GAP period means that all the nodes in the cluster-tree are inactive. Considering that the beacon interval (*BI*) must be large enough to ensure that all superframe durations (SD_i) can be scheduled and that there is also some available spare time during BI, the GAP period is defined based on the following protocol constraint:

$$BI = GAP + \sum_{i=1}^N SD_i, \quad (5.4)$$

Therefore, the GAP period is defined as the time period within the beacon interval during which no cluster is active in the cluster-tree network. Thus, during the GAP period, any RP node can be activated without generating any interference with the cluster-tree network.

In turn, the LAP period means that it may exist some active clusters in the cluster-tree network, but that those clusters are not in the neighbourhood of the RP node. Therefore, a special care must be taken of when activating RP nodes, as these nodes may generate interferences with the cluster-tree network. Unlike the GAP period, which is the same for all RP nodes, the LAP period is dependent of the clusters in the neighbourhood of the RP nodes. For this reason, considering the ARound path and the RP nodes, the ARound-act algorithm avoids the active periods of the clusters and of their neighbours in the cluster-tree network. In this way, an RP node that becomes active to perform the ARound communication does not generate interferences with the remaining clusters that are active in the cluster-tree network.

Thus, considering both the cluster-tree scheduling and the nodes that belong to the ARound path (source, destination and RP nodes), the target of the ARound-act algorithm is to define the activation instants for each of these nodes. For each ARound node it will be defined a transmission time (ARound Tx time) and a reception time (ARound Rx Time), except for the source and destination nodes that will only have, respectively, a defined transmission time and reception time.

When the ARound-act algorithm (Algorithm 4) is launched, the PAN coordinator starts exploring the clusters' active period scheduling (line 3 in Algorithm 4), in order to search the first *ARound activity period* (GAP or LAP, but preferably GAP) that is larger or equal to the required period to allocate to the source node (as transmission node) and to the next RP node (as reception node) of the ARound path. After, the algorithm searches for other *activity period* for this same RP node (now, as transmission node) and to the next RP node (as reception node). This procedure is recursively performed until reaching the destination node (lines 2-14 in Algorithm 4). Whenever the ARound-act algorithm is not able to find any existing ARound activity periods to allocate to all RP nodes (lines 13 and 14 in Algorithm 4), the algorithm fails and the next higher layer is notified. Case all RP nodes are successfully allocated, the ARound-act algorithm defines a set of *ARound_Offsets*, which correspond to the activation time for each RP node (lines 5-7 and 15 in Algorithm 4). As a consequence, each RP node will be awake at its respective *ARound_Offset* time to perform the ARound communication. This communication is then performed without contention, node-to-node, from the source node until reaching the destination node.

In order to reduce the configuration overhead, the ARound path is automatically maintained during a time defined during the configuration phase. After this period, the ARound path must be terminated, unless a new request for message stream communication is performed.

5.4.4 ARound Protocol Mechanisms

In this subsection, we present a set of protocol mechanisms that are required to establish the ARound communication. After running the ARound-def and ARound-act algorithms, the PAN coordinator configures and synchronises all the involved nodes (*ARound negotiation phase*). All these steps can be implemented without modifying the IEEE 802.15.4 standard. In fact, all ARound protocol mechanisms use standard MAC command frames (direct and indirect communication).

Algorithm 4: *Alternative-Path Activation (ARound-act) Algorithm.*

```

input :  $MStream_m$ : [ $SRC_m, DEST_m$ ];
         $ARound\_Path_m$ ; // Shortest inter-cluster path
         $cluster2cluster_m$ ; // Set of pairs of ARound clusters for  $ARound\_Path_m$ 
         $RPnodes$ ; // Set of Repeater nodes
         $Scheduling$ ; // Set of cluster IDs of the cluster-tree scheduling
output:  $ARound\_Offsets$ ; // Set of Offsets for repeater nodes

1 Base =  $Scheduling(initial)$ ; // Receive the ID of the first scheduled
  cluster
2 for (each pair [ $RP_i \rightarrow RP_j$ ]  $\in RPnodes$ ) do
3   while ( $Base < Scheduling(end)$ ) do
4     if ( $get\_Cluster(RP_i)$  and  $get\_Cluster(RP_j)$  are not overlapping with Base) then
5        $ARound\_Offsets_{i,j} = get\_Offset(Base)$ ;
6        $RP_i(SENT) = get\_Offset(Base)$ ;
7        $RP_j(RECEIVER) = get\_Offset(Base)$ ;
8       Base++;
9       break;
10    else
11      Base++;
12  // If reaching the end of the cluster-tree scheduling.
13  if ( $Base == Scheduling(end)$ ) then
14    return (Not possible to allocate all repeater nodes);

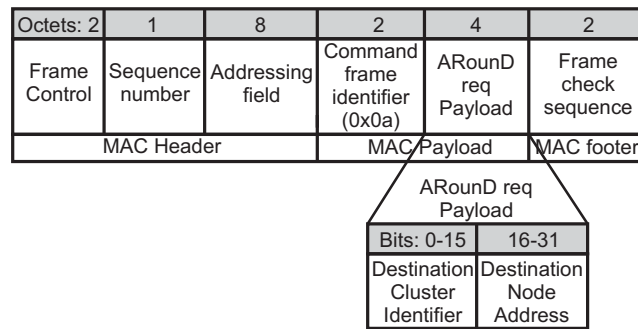
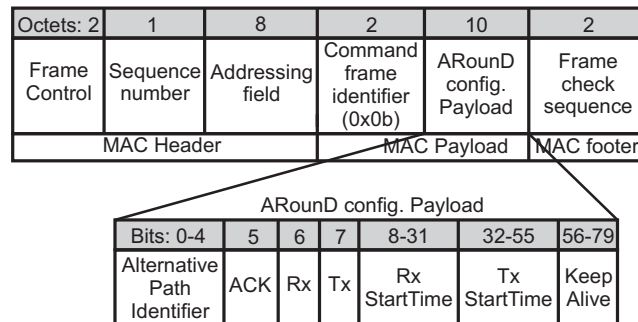
15 return ( $ARound\_Offsets$ );

```

5.4.4.1 Synchronisation and Configuration of the involved nodes

The setup of a new ARound communication path is triggered by the source node using an *ARound request command* frame (ARound req). The *ARound req* frame is transmitted during the CAP period towards the PAN coordinator, using the communication mechanisms defined by the IEEE 802.15.4 standard. Figure 5.9 illustrates the format of this frame. The MAC header fields contain the frame control, sequence number and addressing field. The *Addressing field* contains the *destination Cluster Identifier* and *destination address* fields, which correspond to PAN coordinator. In turn, the *source Cluster Identifier* and *source address* fields correspond to the source node that is requesting the ARound communication path. The 16-bit short address is used as addressing type. In order to identify the *ARound req* frame, the value of the *command frame identifier* field is defined as *0x0a*, which is not used by IEEE 802.15.4. The *ARound req payload* identifies the cluster and address of the destination node.

Within this context, after receiving an *ARound req* frame and performing the ARound algorithms, the PAN coordinator uses an *ARound configuration command* frame (ARound conf) to implement the communication mechanisms required to establish/maintain/close an ARound communication path. Figure 5.10 illustrates the format of this frame. The *Addressing field* contains the *destination Cluster Identifier* and *destination address* fields, which correspond to the RP node to be configured. In turn, the *source Cluster Identifier* and *source address* fields correspond to the PAN coordinator. In order to identify the *ARound conf* frame, the value of the *command frame identifier* field is defined as *0x0b*, which is not used by IEEE 802.15.4.

Figure 5.9: Format of the *ARound-req* frame.Figure 5.10: Format of the *ARound-conf* frame.

The *ARound configuration payload* contains the configuration fields of the RP node specified in the addressing field. The *Alternative path identifier* field identifies the alternative path to be configured, which allows to set up to 32 message streams at the same time (reasonable amount for real WSN applications). The *ACK* field is set to 1 if the *ARound* communication supports acknowledgements; otherwise, this field is set to 0. The *Tx* and *Rx* fields are set to 1 if the involved node is a receiver or/and transmitter, respectively; otherwise, these fields should be set to 0. Note that, if the *Rx* field is set to 1, the *Rx StartTime* field should be configured to the time (*MS_Offset*) when the node will wake up in receiver mode. In turn, if the *Tx* field is set to 1, the *Tx StartTime* field should be configured to the time (*MS_Offset*) when the node will wake up to send data (transmitter mode). In addition, the *Keep alive* field specifies the period (*seconds*) during which the alternative path should be kept active. The PAN coordinator will send the *ARound conf* frames along the cluster coordinators until reaching all the involved nodes, using indirect communication mechanisms.

After receiving the *ARound conf* frame, the involved node sends an acknowledgement frame to inform the PAN coordinator, using the *ARound ack command* frame. Figure 5.11 illustrates the format of this frame. In order to identify the *ARound ack* frame, it is used the value *0x0c* in the *command frame identifier* field, which is not used by the IEEE 802.15.4 standard. The *Addressing field*, *destination Cluster Identifier* and *destination address* fields correspond to the PAN coordinator. In turn, the *source Cluster Identifier* and *source address* fields correspond to the configured node. The 16-bit short address is used for addressing purposes.

The *ARound ACK payload* field is composed of the *alternative path identify* field (5-bits) to identify the alternative path and the *ACK type* field to specify the *acknowledgement frame type*. Table 5.1 summarizes the *ACK frame* types used in the *ARound* communication scheme.

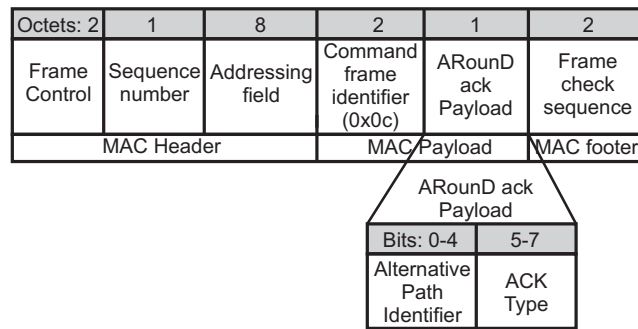
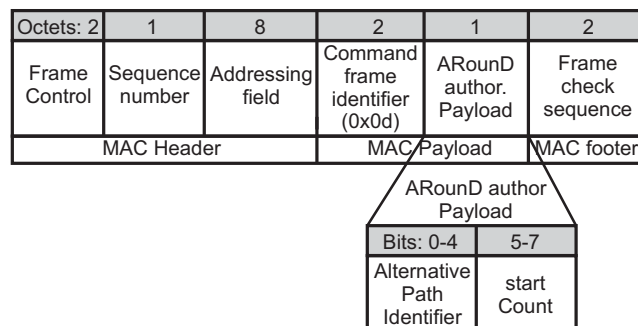
Figure 5.11: Format of the *ARound* acknowledgement command frame.

Table 5.1: ACK frame types for the ARound communication scheme.

Frame Type Value	Description
000	ACK for an ARound configuration frame
001	ACK for an ARound closing frame
010	ACK for an ARound hello frame
011-111	Reserved

After receiving all the acknowledgements from the involved nodes (source, destination and RP nodes), the PAN coordinator will authorise the start of the new ARound communication using an *ARound* *authorisation* command frame (ARound auth). Whenever the PAN coordinator does not receive all the required acknowledgements, it will retry to configure the missing nodes for three times. If the configuration does not succeed after these three attempts, the ARound path will not be created.

The *ARound* *auth* frame is sent to the involved nodes using indirect communication mechanisms. Figure 5.12 illustrates the format of this frame. The value *0x0d* is used for the *command frame identifier* field, which is not used by the standard. Similarly to the configuration procedure, the *Addressing field* contains the *destination Cluster Identifier* and *destination address* fields, which corresponds to the node to be authorized. In turn, the *source Cluster Identifier* and *source address* fields correspond to the PAN coordinator.

Figure 5.12: Format of the *ARound* *auth* frame.

The *ARound* *auth* *payload* field is composed of the *alternative path identifier* and *start count* fields. While the *alternative path identifier* field identifies the alternative path, the *start count* field defines the number of beacon intervals after which the new ARound communication will start.

Figure 5.13 illustrates the message flow diagram for the *ARound negotiation phase*. As it can be seen, the *ARound conf* frame is propagated through the coordinators along the network using indirect communication and all the involved nodes will be configured with the *ARound* parameters. These nodes save the necessary information and send acknowledgement frames towards the PAN coordinator. After receiving the acknowledgement frames from all involved nodes, the PAN coordinator will send an *ARound auth* frame to authorise the start of this communication. After this, the new *ARound* path will be launched at $BI_{node} \times Start_Count + MS_Offset_{node}$ time.

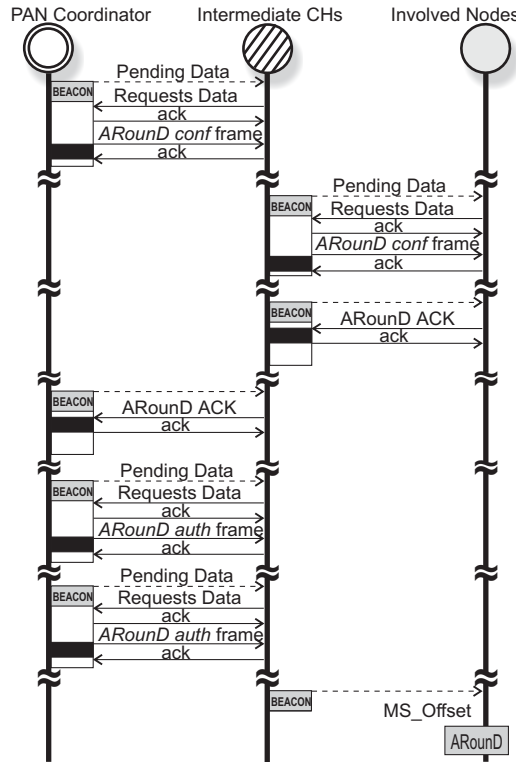


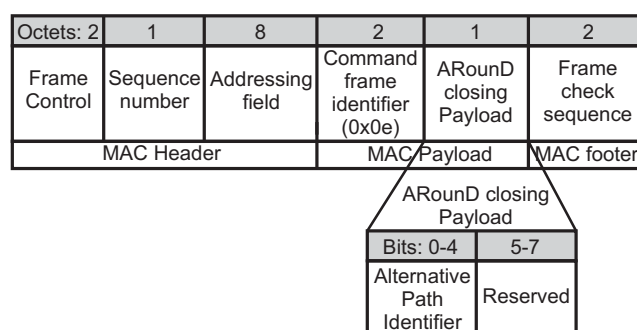
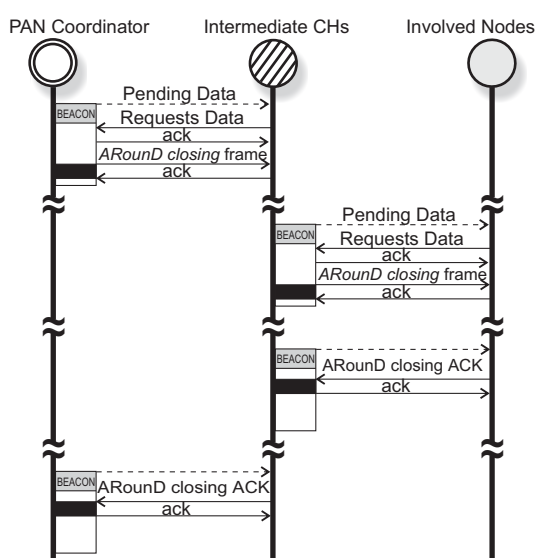
Figure 5.13: Message flow diagram for the *ARound negotiation phase*.

5.4.4.2 Closing the *ARound* Communication

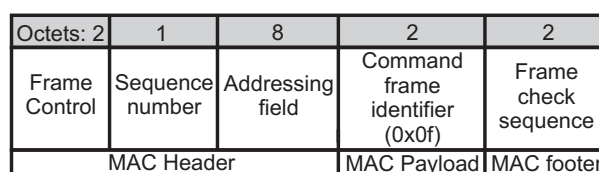
As presented in the previous subsection, the *ARound* communication is configured to work during a given time, which is specified by the *keep alive* field. Nevertheless, the *ARound* communication also provides an explicit mechanism to close an on-going alternative path.

The *ARound communication closing* mechanism is launched by the PAN coordinator, which sends an *ARound closing* frame to all the involved nodes. Figure 5.14 illustrates the format of this frame. After receiving this frame, the involved nodes immediately finish the *ARound* communication. Finally, each involved node sends to PAN coordinator an *ARound ack* frame (ACK type field set to 001).

Figure 5.15 illustrates the message flow diagram for the *communication closing* mechanism, highlighting that every node sends an *ack* frame to the PAN coordinator to end the *ARound* communication.

Figure 5.14: Format of the *ARound* closing command frame.Figure 5.15: Message flow diagram for the *ARound* closing mechanism.

Finally, the *ARound* communication also enables the PAN coordinator to verify if all the involved nodes are still alive, using an *ARound hello* frame. Figure 5.16 illustrates the format of this frame. Note that the MAC payload is composed of the *command frame identifier* field only (with value set to 0x0f, which is not used by the IEEE 802.15.4), in order to identify this frame. The PAN coordinator may query a specific node to confirm if it is still alive. This node replies with an *ARound ack* frame (*ACK type* field set to 010).

Figure 5.16: Format of the *ARound hello* command frame.

5.5 Simulation Assessment of the ARound Communication Scheme

In the previous sections, a set of communication mechanisms (ARound) were proposed, enabling the setting-up of new communication paths upon a cluster-tree wireless sensor network. These new paths use available communication resources, without interfering with the current cluster-tree communication.

In this section, we present a simulation assessment of the proposed *ARound* communication scheme. The rationale behind this assessment is to compare the behaviour of a specific message stream when its messages are transferred using the ARound communication scheme, or using the standard cluster-tree routing scheme. We consider that the cluster-tree network is already been used to support a set of background message streams. That is, there is a set of sensor nodes sending periodic messages towards the PAN coordinator through the cluster-tree network.

For this simulation assessment, it was used the Castalia Simulator [117]. Castalia¹ is an open-source discrete event simulator for wireless sensor networks (WSN), Body Area Networks (BAN) and general low-power embedded networks. It was developed at National ICT Australia (NICTA) and is based on the OMNeT++ platform widely used by researchers and developers to test their protocols using realistic wireless channel and radio models. According to [87], Castalia has become a very popular simulator as it was developed for research. Castalia implements an advanced wireless channel model based on empirically measured data. Also, the simulator provides radio models based on real low-power communication radios. Moreover, important features to simulate wireless sensor networks are available, such as: realistic node behaviour, node *clock drift*, and energy consumption models.

Castalia provides an IEEE 802.15.4 model. However, this model is quite limited. Basically, it only implements the CSMA-CA functionality and a beacon-enabled star topology, including an association procedure, direct data transfer mode, and GTS communication. Therefore, we developed the **ARound simulation model**, which includes a series of functionalities, such as: cluster-tree formation procedure, network scheduling, hierarchical addressing scheme, direct and indirect data communication, collision domain definition, data communication to the sink node (PAN coordinator), the ARound communication mechanisms and source-to-destination message streams using both cluster-tree and ARound routing algorithms.

5.5.1 Simulation Environment

In order to assess the ARound communication scheme, we defined a simulation environment with size to $200\text{ m} \times 200\text{ m}$, containing 503 sensor nodes (one PAN coordinator, one pair of source and destination nodes, plus 500 general purpose nodes). In this assessment, we adopted the radio model CC2420², which is compliant with IEEE 802.15.4. Moreover, a linear energy model provided by Castalia was used for simulations. The initial energy for all nodes

¹The Castalia Simulator for Wireless Sensor Network: <https://castalia.forge.nicta.com.au>.

²Texas Instruments / Chipcon CC2420 Datasheet: <http://www.ti.com/product/CC2420/technicaldocuments>.

was set to 18720 Joules, which is the typical energy for two AA batteries. As radio propagation model, we adopted the *unit disc* model, where the range of the disk was defined to 55m. In addition, we used an interference model where concurrently transmissions generate collisions at the receiver.

For the nodes' deployment, we set node 0 as the PAN coordinator and nodes 1 and 2 as the source and destination nodes for the peer-to-peer message stream, respectively. These nodes are statically deployed in the communication environment. The PAN coordinator is located in the central position of the field (100 m, 100 m). For nodes 1 and 2, we defined two different scenarios, in order to avoid biased assessments. Firstly, nodes 1 and 2 were deployed in positions (30 m, 30 m) and (170 m, 30 m), in adjacent quadrants (*Scenario 1*). This configuration was used to have a balanced deployment. Secondly, nodes 1 and 2 were deployed in positions (30 m, 30 m) and (170 m, 170 m), in opposite quadrants (*Scenario 2*). This configuration was used to configure a worst-case scenario, where the source and destination are geographically distant and the PAN coordinator is located between the two nodes, forcing the peer-to-peer communication through more congested areas of the network.

The other 500 general-purpose nodes were randomly deployed in the environment. Basically, five different physical topologies were generated for setting-up the position of these nodes, which were used for the assessment of the ARound communication mechanisms. Thus, we have 10 different physical deployments: five physical topologies for *Scenario 1* and five physical topologies for *Scenario 2*. For each of the generated physical topologies, 10 simulations were run with different sets of random variables. The results presented along this section correspond to the average of results obtained from these simulations. Note that, basically, the difference between the five physical topologies for the *Scenario 1* and *Scenario 2* is the position of the node 2. Figure 5.17 illustrates one of 5 physical topologies considering the source and destination nodes located in adjacent quadrants (*Scenario 1*), while Figure 5.18 illustrates one of 5 physical topologies considering the source and destination nodes located in opposite quadrants (*Scenario 2*).

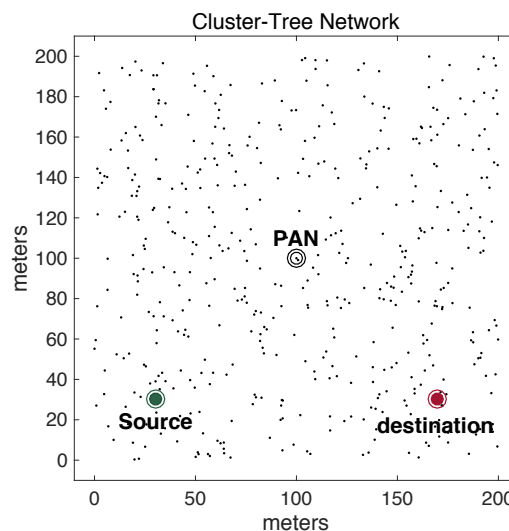


Figure 5.17: Configuration of the *Scenario 1*: nodes 1 and 2 in adjacent quadrants.

The cluster-tree formation is based on the IEEE 802.15.4/ZigBee standard. The network formation is started by the PAN coordinator, which is responsible to build its own cluster and

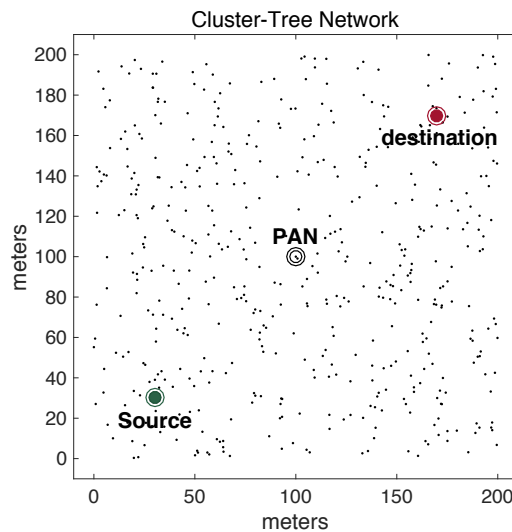


Figure 5.18: Configuration of the *Scenario 2*: nodes 1 and 2 in opposite quadrants.

acts as cluster-head. This cluster is the first of the cluster-tree network and corresponds to depth 0. For the *addressing scheme*, we used the ZigBee-based *hierarchical addressing scheme*, which allows a hierarchical tree routing according to address blocks.

For the clusters' active period scheduling, we used a simple time division scheme. These periods are scheduled according to a *top-down* scheme, starting with the PAN coordinator. Then, the clusters of the next level are scheduled, and so on, until reaching the maximum depth of the tree. The PAN coordinator performs the cluster-tree scheduling and checks if the current cluster-tree configuration is schedulable.

Table 5.2 summarizes the most important configuration parameters used in the simulations and pointed out in this section.

5.5.1.1 Characterisation of Data Traffic

After finishing the cluster-tree network formation, sensor nodes start the *data communication phase*. In the simulations, two types of data communication were defined, as follows:

- *background data traffic*: 500 general-purpose sensor nodes sending periodic data towards the sink node (PAN coordinator).
- *source-to-destination data traffic*: a specific source node sending periodic data to a specific destination node.

In the simulations, both the PAN coordinator and the source and destination nodes do not generate any background data traffic. Thus, there are 500 sensor nodes generating background data traffic. Each sensor node generates a maximum of 1.000 data frames. Considering this setup, the background traffic load corresponds to a sequence of 500.000 data frames sent by 500 randomly located sensor nodes to the PAN coordinator (sink node). Importantly, the cluster-heads do not perform any data aggregation or data fusion technique,

Table 5.2: Simulation configuration.

Definition	Standard Value
Environment size	200 m x 200 m
Number of sensor nodes	503
Nodes sending Background Data	500
Radio model	Chipcon CC2420
Initial energy (per node)	18.720 Joules
Simulation time (each experiment)	85.000 <i>seconds</i>
Number of Background Data Frames (per node)	1.000
Background Data Rate	from 1 pkt/60s up to 1 pkt/40s
Number of Source-to-Destination Data Frames (Node 1)	10.000
Source-to-destination Data Rate	1 pkt/4s and 1 pkt/8s
physical Data Rate	250 kbps
<i>aBaseSlotDuration</i>	60
<i>aNumSuperframeSlots</i>	16
<i>aUnitBackoffPeriod</i>	20
<i>BeaconOrder</i>	9
<i>superframeOrder</i>	ranging from 0 to 4
<i>macMaxCSMABackoffs</i>	4
<i>macMaxFrameRetries</i>	2

which implies that all background traffic is forwarded towards the sink node. In future works, these techniques will be also implemented in the *ARound simulation model*.

In order to select an adequate value for the background data rate, we performed a set of experiments to evaluate the average number of queued packets at the cluster-head nodes, using different background data rates. The intention was to create an active but not congested simulation environment. Figure 5.19 illustrates the average number of packets in the MAC buffers of the cluster-heads at different depths of the cluster-tree network (PAN coordinator corresponds to depth 0).

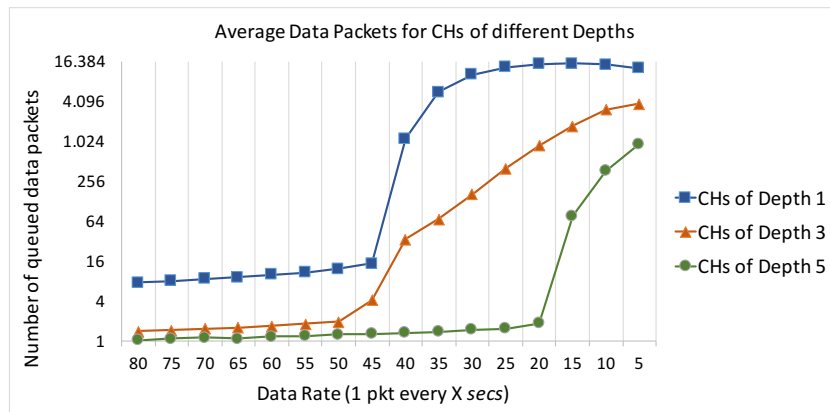


Figure 5.19: Average number of queued data packets in MAC buffers.

Note that, from a quite small data rate of 1 packet every 45 *seconds*, the number of queued packets in the buffers of the cluster-heads of depth 1 exponentially increases, meaning that

these cluster-heads will be quickly facing queue overflows. For a data rate of 1 packet every 15 *seconds*, the scenario is much worse, as it will result in a fully congested environment, where a huge number of data packets are automatically dropped by the network. Within this context, we selected for the background data traffic the following data rates: from 1 packet every 60 *seconds* up to 1 packet every 40 *seconds*, which may correspond to typical monitoring applications.

The source-to-destination data traffic is set for node 1, which sends 10.000 data frames to node 2 according to the following data rate configurations: 1 message every 4 *seconds* (1 pkt/4s) and 1 message every 8 *seconds* (1 pkt/8s). These are typical values that can be found for the transmission of video streams in wireless visual sensor networks [131]. This peer-to-peer message stream is started after a simulation time of 100 *seconds*.

5.5.1.2 Performance Metrics

The aim of this simulation assessment is to evaluate the behaviour of the ARound communication scheme when compared to the use of cluster-tree routing and its impact over the background communication traffic. For the sake of convenience, we adopted the name *Standard Approach* for the cluster-tree routing and *ARound Approach* for the ARound communication routing. The following performance metrics were used for the analysis of the network behaviour:

- *End-to-end Delay*: time interval between the data frame generation at the application layer of the source node and its reception at the application layer of destination node.
- *Packet Loss Rate*: the percentage of packets lost during the communication, considering the number of data packets successfully received at the destination node and the number of packets generated by the source node.
- *Energy Consumption*: average energy consumption of the overall network.

5.5.2 Results and Discussion

Table 5.3 shows results from the network formation process for all the considered topologies (*Scenarios 1* and *2*), concerning the average number of generated clusters during the cluster-tree formation, the average maximum depth of the cluster-tree network and the average number of children per cluster. Importantly, all the communication environment was fully covered, meaning that all nodes were associated with a particular cluster-head (no orphan nodes).

Table 5.3: Information about the network formation.

Information	Value
Average number of clusters	59
Average maximum depth	5
Average number of children per cluster	8

Regarding the source-to-destination communication, Figure 5.20 presents some information about nodes 1 (source) and 2 (destination), and the source-to-destination message stream. Figure 5.20a shows the average depth for nodes 1 and 2 in the cluster-tree network for *Scenarios 1* and 2. As it can be seen, nodes 1 and 2 are commonly located in a cluster of depth 3 in the cluster-tree topology for *Scenario 1*. Otherwise, nodes 1 and 2 are commonly located in a cluster of depth 4 in *Scenario 2*. In turn, Figure 5.20b shows the average number of hops from the source node to destination node for both *Scenarios*, when using the cluster-tree routing versus the ARound path.

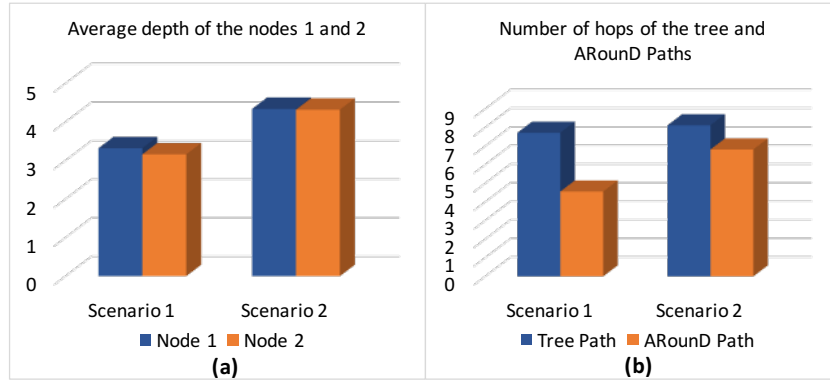


Figure 5.20: Information about source-to-destination message stream, considering the scenarios 1 and 2: (a) Average depth of nodes 1 and 2; (b) Number of hops for the cluster-tree and ARound paths.

Note that, as expected in *Scenario 1*, the ARound path is in average smaller than the cluster-tree path. This is basically due to the ARound approach to select the shortest inter-cluster path between the source and destination nodes. For *Scenario 2*, the average difference between the cluster-tree and ARound paths is smaller. In fact, this configuration can be considered as a worst-case scenario for the ARound scheme, where the PAN coordinator is located between source and destination nodes. However, even though cluster-tree and ARound paths are equivalent, the main advantage of ARound is to prioritise leaf nodes to build the path, allowing to save energy of cluster-tree nodes. Figure 5.21 illustrates one of the physical topologies for *Scenarios 1* and 2. Note that both ARound paths are composed mostly of leaf nodes.

Although the ARound paths are usually smaller than cluster-tree paths, the main target of these simulations is to evaluate the performance of the message stream using the ARound scheme compared to the standard cluster-tree routing, regarding end-to-end delays and successfully received rate for the data messages.

Within this context, Figure 5.22 shows the average end-to-end delay of the source-to-destination message stream when using the ARound mechanisms, when compared to the standard cluster-tree routing for simulation *Scenario 1* (five physical topologies, under different background and source-to-destination traffic loads). The *ARound approach* provides a smaller end-to-end delay compared to the IEEE 802.15.4 standard tree routing (*Standard approach*) for all background and source-to-destination data rates. This is due not just to the shortest inter-cluster path between the source and destination nodes; the main reason is that the path is scheduled during just one beacon interval. In turn, the standard cluster-tree approach spans over several beacon intervals to transmit data frames,

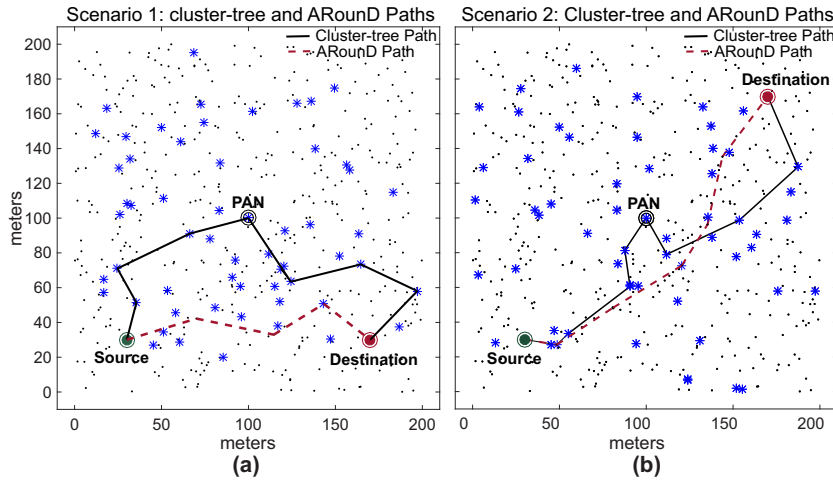


Figure 5.21: Cluster-tree and ARoundD paths for one of physical topologies: (a) *Scenario 1*; (b) *Scenario 2*.

which increases the end-to-end delay of the messages. This is one of the major disadvantages of cluster-tree path scheduling, where either a top-down or a bottom-up approach is used for scheduling the traffic (but not both). It is well-known that bottom-up scheduling approaches favour traffic scheduling from sensors (source node) to the PAN coordinator (sink node), while top-down approaches favour the opposite traffic direction [53]. Moreover, the contention and transmission attempts of the CSMA-CA algorithm will also impact the end-to-end delay in more congested areas of the wide-scale cluster-tree (near the PAN coordinator) [99].

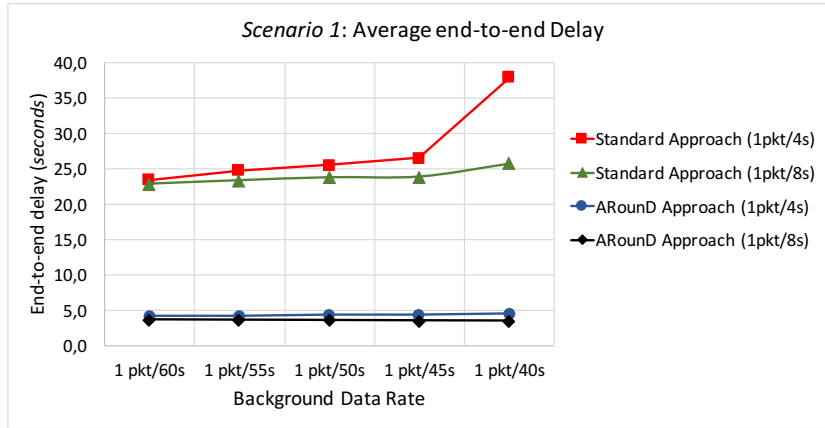


Figure 5.22: End-to-end Delay for ARoundD Approach vs. Standard Approach considering *Scenario 1*.

It can be also observed that, as the background data rate increases, the average end-to-end delay for the standard cluster-tree approach also increases. This is mainly due to the accumulation of queued data packets in the cluster-heads, causing further delays to the newly arriving packets. However, as the ARoundD scheme defines an alternative path using RP nodes (which are mainly leaf nodes), the end-to-end delay for the source-to-destination traffic is just slightly affected. As it was expectable, the average end-to-end delay is highly dependent on the background traffic load for the standard cluster-tree approach, as the

source-to-destination traffic is sharing the same MAC queues as the background traffic. On the other hand, in the ARound scheme, as the source-to-destination traffic almost does not share RP nodes with the background traffic, the average end-to-end delay keeps constant for the different data rates.

Figure 5.23 shows the average end-to-end delay of the source-to-destination message stream when using the ARound mechanisms, when compared to the standard cluster-tree routing for the simulation *Scenario 2* (five physical topologies, under different background and source-to-destination traffic loads). Also, the *ARound approach* provides a smaller end-to-end delay compared to the *Standard approach* for all background and source-to-destination data rates.

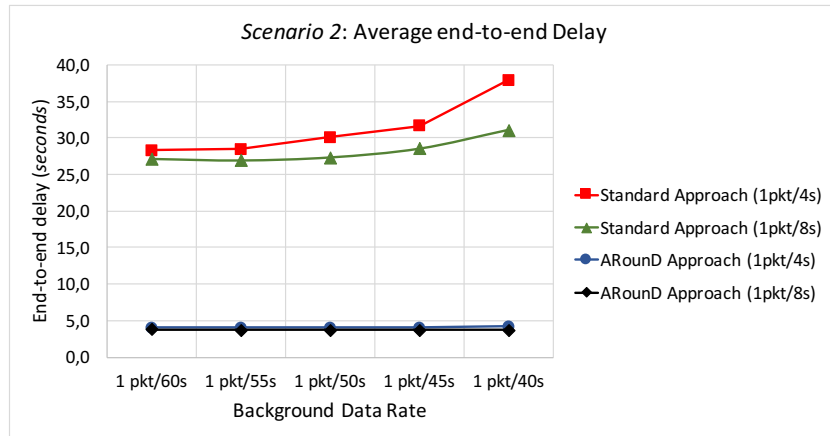


Figure 5.23: End-to-end Delay for ARound Approach vs. Standard Approach considering *Scenario 2*.

Figure 5.24 illustrates the packet loss rate due to buffer overflows for the ARound approach when compared to the standard approach for the *Scenario 1*, while Figure 5.25 illustrates the packet loss rate for the *Scenario 2*. It can be observed that the ARound scheme successfully delivered all the source-to-destination messages, while the standard cluster-tree path is prone to a high number of packet losses due to network congestion issues. Additionally, as the ARound communication scheme defines paths during the inactivity period of the clusters, it does not need to contend for the channel access, which avoids collisions and the resulting packet losses. Although providing the GTS mechanism to transmit time-sensitive packets without contention, the main weakness of this mechanism is its limited number of available slots. Note also that, similarly to the case of average end-to-end delay, the ARound communication scheme is able to keep the same packet loss rate for the different source-to-destination data rates.

An important issue is also the assessment of the impact of the *source-to-destination traffic* over the *background traffic*, when compared to sending that traffic through the standard cluster-tree approach. For this, a set of simulations were performed considering only background data traffic, in order to capture its behaviour without the source-to-destination traffic. After, the behaviour of the background data traffic was analysed, considering the addition of source-to-destination data traffic for different background data rates. Figure 5.26 illustrates the results obtained from this assessment for *Scenario 1*, for both ARound and Standard approaches. Basically, it can be observed that the source-to-destination traffic has a negligible impact over the background traffic. This is due to

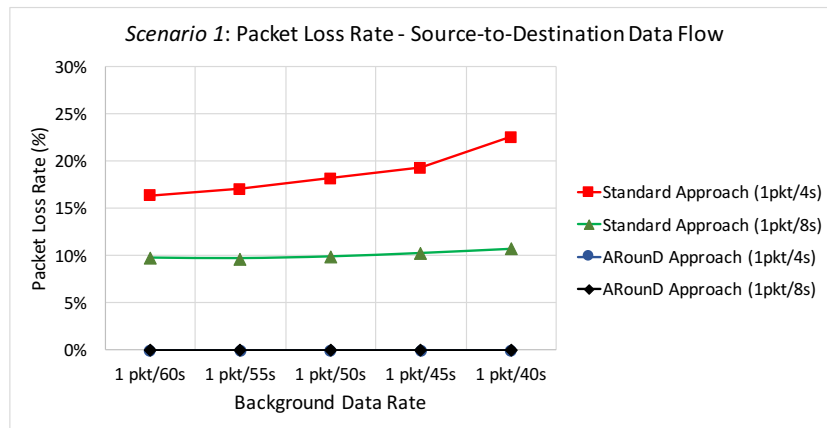


Figure 5.24: Packet Loss Rate for ARound Approach vs. Standard Approach for *Scenario 1*.

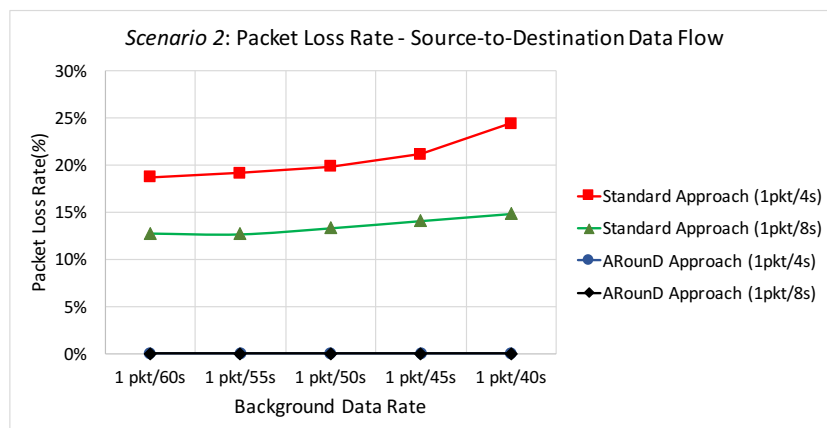


Figure 5.25: Packet Loss Rate for ARound Approach vs. Standard Approach for *Scenario 2*.

the considered non congested environment. However, increasing the graph scale, it can be observed that the Standard approach has an impact greater than that caused by the ARound communication scheme. Note that when increasing the background traffic data rate, the source-to-destination traffic using the Standard approach has a visible impact over the background traffic, while the impact from the ARound traffic keeps the negligible impact. In fact, as the standard approach also uses the cluster-tree path to transfer source-to-destination traffic, the cluster-tree network has more packets contending for the communication path, increasing the packet loss rate. The similar behaviour was also observed for *Scenario 2*.

It was also assessed the impact of *source-to-destination data traffic* upon the average end-to-end delay of *background data traffic*. As it can be seen in Figure 5.27, the ARound communication traffic does not interfere with the background traffic. The main reason is that ARound defines an alternative path to transmit source-to-destination traffic. Conversely, as the Standard approach uses the cluster-tree path to transmit its *source-to-destination data traffic*, the number of packets in the MAC buffers tends to increase, impacting the background traffic end-to-end delay. Also, the same behaviour was observed for *Scenario 2*.

Finally, the overall energy consumption of the network when using the proposed ARound communication scheme was also assessed. For this, and differently to the previous simulation

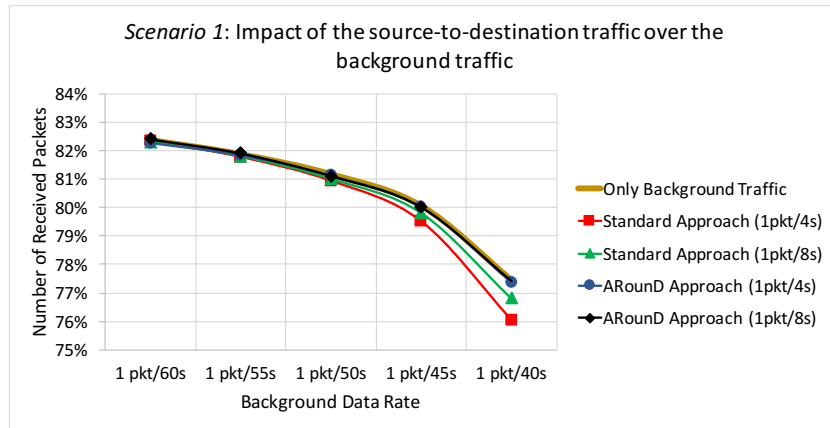


Figure 5.26: Impact of the *source-to-destination traffic* over the *background traffic* (Scenario 1).

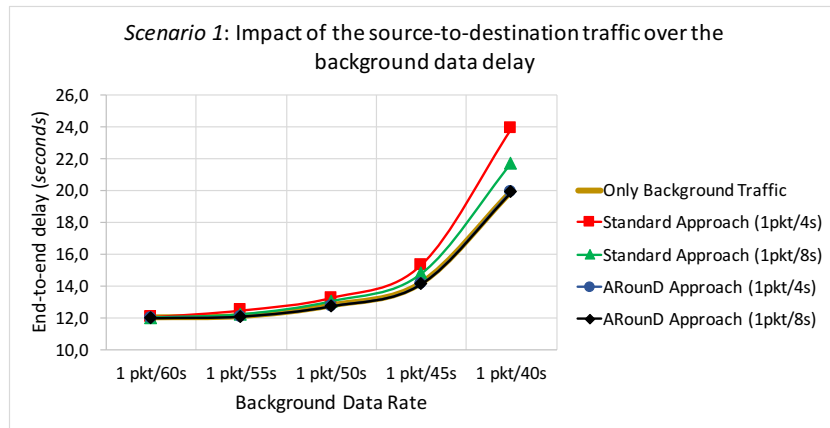


Figure 5.27: Impact of the *source-to-destination data traffic* (ARound and the Cluster-tree approaches) over the *background data traffic* delay, considering the Scenario 1.

scenario, 20 different topologies were analysed, with the PAN coordinator deployed in the center and all other nodes randomly deployed (including nodes 1 and 2). The background traffic data rate was set to 1 pkt/50s, while source-to-destination traffic data rate was defined to 1 pkt/4s. In order to assess the overall energy consumption of the network, nodes were organized in the following groups: PAN coordinator, cluster-heads (CHs) and ordinary sensor nodes (Nodes). Figure 5.28 illustrates the overall energy consumption for both the ARound and Standard communication schemes. It can be observed that the ARound communication scheme saves energy for all node groups. For the PAN coordinator, the ARound approach saves about 8,88% when compared to the Standard approach. For the cluster-heads, it saves about 8,44% of the energy consumption. Importantly, as the number of source-to-destination messages increases, the difference between the energy consumption of the node groups using the ARound and Standard schemes can potentially be greater. Therefore, it can be stated that using the ARound communication scheme will improve the energy consumption of the overall cluster-tree network and, therefore, it will increase its lifetime.

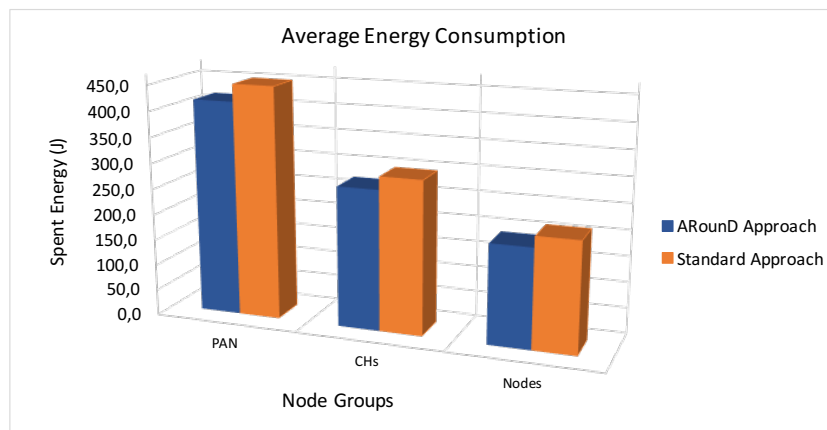


Figure 5.28: Energy Consumption for the different node groups.

5.6 Conclusions

In this paper, we present the ARound (*Alternative-Route Definition*) communication scheme, that enables the definition of alternative paths for peer-to-peer message streams in IEEE 802.15.4/ZigBee-based cluster-tree WSNs. The main target of this communication scheme is to define an alternative inter-cluster path using nodes during their inactive periods, in order to reduce the interference with the cluster-tree network traffic. As a consequence, the ARound communication scheme avoids transferring messages through the underlying cluster-tree paths, being able to improve several network metrics such as: network congestion, overall energy consumption and end-to-end communication delays.

Simulations results highlight that the use of the ARound communication scheme can significantly decrease the end-to-end communication delay for source-to-destination data traffic, because it defines the shortest inter-cluster path between source and destination nodes and it is able to schedule this path in just one beacon interval. Furthermore, as source-to-destination messages do not contend for the wireless channel access with other messages, it avoids additional delays and dropped packets due to collisions. It is also shown that the ARound communication scheme has just a negligible impact upon the cluster-tree communication and can be used to support the transfer of messages, guaranteeing limited QoS requirements such as bandwidth and softly-bounded delays. Moreover, the proposed scheme saves energy for all node groups, which will potentially increase the lifetime of the cluster-tree network.

5.6.1 Future Considerations

As future considerations, other metrics will also be used to define alternative paths, such as link quality and energy capability. Besides, new functionalities will be also added to the ARound simulation model, such as: new scheduling approaches and aggregation and data fusion mechanisms.

Conclusions and Future Work

In this chapter, the major conclusions of the research work proposed in this Thesis and some directions for future works are presented.

6.1 Thesis Summary

This Thesis addresses some relevant research issues related to IEEE 802.15.4/ZigBee cluster-tree WSNs applied to wide-scale deployments. Basically, the research work presented in this Thesis focuses on the design of new efficient communication mechanisms to improve source-to-destination message stream communication in wide-scale cluster-tree WSNs.

Wireless Sensor Networks (WSNs) have become an important research topic, since this type of network has been identified as an attractive technology to deploy wide-scale applications, such as those found in: industrial automation, military applications, precision agriculture and environmental monitoring. In Chapter 2, we present some considerations about wide-scale WSNs, by considering a state-of-the-art study about standards and routing protocols for WSNs. Within this context, the IEEE 802.15.4/ZigBee set of standards is one of the most used technologies to build WSN-based applications. This set of standards provides the cluster-tree topology, which is pointed out as a suitable topology for building wide-scale WSN applications. Although the IEEE 802.15.4 provides the cluster-tree topology, this standard does not explicitly define mechanisms to deal with cluster-tree networks. In turn, ZigBee defines a set of rules to build cluster-tree networks, related to the network formation, addressing and routing mechanisms.

Cluster-tree networks present a series of advantages to build wide-scale applications, mainly related to the beacon mode operation, such as: energy-efficiency and duty-cycle operation, timing synchronisation, collision-free communication, distributed network control and reduced network contention. Moreover, the continuous advances in MEMS, processing, storing and wireless radio technologies allow the design of low-power and low-cost sensor devices, enabling the deployment of cluster-tree networks with a large number of sensor nodes for wide-scale environments.

However, this type of topologies still presents several open research issues, that motivate the design of new approaches and protocols to efficiently deal with wide-scale cluster-tree networks and their main operation and communication mechanisms. We identified four research problems, as follows: cluster-tree network formation, beacon frame scheduling, network parameter configuration, and efficient and alternative communication schemes. In

this Thesis, we have focused on providing new mechanisms to deal with network parameter configuration and alternative communication issues.

Several network parameters have direct impact upon the performance of IEEE 802.15.4 MAC protocol. This way, the defined values for these parameters may influence the network operation and the expected results. Moreover, these parameters are also related to several known problems in cluster-tree networks, such as: high end-to-end delays, congestion and discarded messages. Therefore, adequate configuration of these parameters may improve the network performance and avoid undesired network operation. In addition, for applications that impose stringent requirements, the definition of these parameters becomes a crucial design issue. These assumptions have motivated the design of new mechanisms to efficiently define the MAC communication parameters for wide-scale cluster-tree WSNs.

Besides that, cluster-tree networks assume a tree-based routing where all communication traffic follows the tree paths (parent-child relationships) toward the PAN coordinator. Despite its simplicity, this type of routing imposes a series of problems, such as: high number of hops along the path, higher end-to-end delays, higher energy consumption of crucial nodes and congestion, mainly near the PAN coordinator (commonly, the sink node). This way, the design of new mechanisms that enable the use of alternative paths is an attractive approach, that may avoid some of these problems and improve the network performance.

Within this context, this Thesis presents two new communication schemes for cluster-tree networks: *Proportional Superframe Duration Allocation* (SDA) schemes and *Alternative-Route Definition* (ARound) communication scheme.

Additionally, the implementation of real-world wide-scale applications involves high cost and time resources, which motivate the use of other approaches to analyse and validate new designs. Thus, simulation approaches have become an attractive method to assess new implementations, before their real-world deployments. However, most part of available simulators do not provide the essential mechanisms to deal with wide-scale cluster-tree networks. For this reason, we propose a simulation model for building and testing the new cluster-tree communication schemes, as those proposed in this Thesis. Thus, Chapter 3 presents the CT-SIM simulation model for IEEE 802.15.4/ZigBee cluster-tree networks. This simulation model implements the main mechanisms for automatically building cluster-tree networks, such as: cluster-tree formation based on several network attributes, hierarchical addressing, beacon scheduling mechanisms, proportional superframe duration allocation schemes, direct and indirect data communication and different message streams models. The CT-SIM uses a modular structure and was developed upon Castalia Simulator, allowing researchers to implement and test new communication protocols.

In Chapter 4, we present the *Proportional Superframe Duration Allocation* (SDA) scheme. We firstly define a set of worst-case boundary equations to model the protocol and timing constraints imposed by cluster-tree networks. The protocol constraint considers the main requirements imposed by IEEE 802.15.4 MAC protocol. The timing constraint provides a boundary equation associated to worst-case response time for transferring data frames. With this, we have presented two proportional superframe duration schemes: Load-SDA and Nodes-SDA schemes. The Load-SDA scheme defines superframe durations, beacon intervals and buffer size values for each cluster-head, considering the message load imposed by its child nodes (including all descendant nodes). The underlying idea of the Load-SDA scheme is to guarantee that each cluster-head has enough resources to support

the monitoring traffic load imposed by the descendant nodes. This scheme is suitable for cluster-tree networks, where the message load imposed by nodes are known. In turn, the Nodes-SDA scheme is used for cluster-tree networks, where the traffic load is unknown. In this case, the number of child nodes is considered (including all descendant nodes), assuming that all nodes generate a similar traffic load. We showed through simulation that the Load-SDA and Nodes-SDA schemes have a better network performance, when compared to a static scheme, which considers the same values of superframe durations, beacon intervals and buffer sizes for all cluster-heads. For this simulation assessment, we considered the following network metrics: number of discarded messages due to buffer overflows, end-to-end communication delay, message loss rate and energy consumption. Therefore, the results presented in this chapter show that the definition of a set of guidelines to adequately allocate communication structures may improve the network performance and avoid some of the problems that may turn difficult the use of cluster-tree networks to support wide-scale applications.

Finally, Chapter 5 presents the *Alternative-Path Definition* (ARound) communication scheme. The underlying idea of the ARound scheme is to define new alternative inter-cluster paths among the cluster (instead of tree paths), using border nodes during their inactive periods. By avoiding the tree paths, the ARound scheme may decrease the congestion of the network and save communication resources, mainly of crucial nodes (cluster-heads). By defining the shortest inter-cluster paths, the ARound may improve end-to-end communication delays and guarantee limited QoS requirements. By using sensor nodes during their inactive periods, the ARound scheme does not interfere with the standard cluster-tree communication. Simulation assessments have shown that the ARound communication scheme may improve the network performance, when compared to standard cluster-tree communication. The ARound scheme decreases the end-to-end communication delay for source-to-destination message streams, guaranteeing bandwidth and softly-bounded delays. Moreover, this scheme saved energy of all nodes, which may potentially increase the network lifetime.

Therefore, the main contributions of this Thesis can be summarised as follows.

- A new simulation model (CT-SIM) for wide-scale cluster-tree networks based on the IEEE 802.15.4/ZigBee standards, that considers their main mechanisms, such as: cluster-tree formation based on several network attributes, hierarchical addressing, beacon scheduling mechanisms, proportional superframe duration allocation schemes, direct and indirect data communication and different message streams models.
- A set of guidelines to adequately allocate superframe durations, beacon intervals, and buffer size values for the cluster-heads of a cluster-tree network, in order to improve the throughput of the monitoring traffic and to avoid typical problems, such as: network congestion, higher end-to-end communication delays and discarded messages due to buffer overflows.
- A new message communication scheme that defines alternative inter-cluster paths to support end-to-end message streams in cluster-tree networks.

6.2 Future Work

The research work presented in this Thesis will be the beginning for a set of future research about Wireless Sensor Networks and wide-scale applications. In the next two years, we intend to address, at least, the following research topics:

- **New modules for CT-SIM simulation model.** Although the CT-SIM encompasses several mechanisms to deal with cluster-tree networks, we intend to implement new mechanisms for the simulation model to allow the design of a series of new application domains, such as: data aggregation and fusion mechanisms, CSMA-CA parameters configuration schemes, new beacon frame scheduling mechanisms and new data communication schemes. Moreover, we intend to make available a beta version (with documentation) soon.
- **To extend the MAC parameter configuration schemes.** This Thesis provides a set of guidelines on how to setup the MAC parameters, in what concerns its communication structures. Following this Thesis, we intend to explore multiple configurations for the CSMA-CA parameters, in order to prioritise the message streams, and to integrate these new message priority mechanisms within the SDA schemes.
- **To design and implement new communication approaches for cluster-tree networks.** The constant need of efficient communication mechanisms for cluster-tree networks has motivated the design of new communication paradigms. Therefore, we intend to advance in this research topic, in order to provide new communication schemes and to adapt existent schemes (as the proposed in [99]). Also, we intend to extend the ARoundD scheme and add new rules and mechanisms that allow the definition of several paths, based on different goals.
- **New paradigms for WSNs.** Finally, we intend to explore and contribute into other WSN paradigms, such as: new communication mechanisms for healthcare applications and Internet of Things.

APPENDIX A

List of Publications

A list of publications produced during this PhD is presented below.

A.1 Submitted Manuscripts to Journals

Erico Leão, Carlos Montez, Ricardo Moraes, Paulo Portugal, and Francisco Vasques. "Alternative Path Communication in Wide-Scale Cluster-Tree Wireless Sensor Networks using Inactive Periods.", *Sensors*, MDPI, 2016. (**UNDER CONSIDERATION**).

Erico Leão, Carlos Montez, Ricardo Moraes, Paulo Portugal, and Francisco Vasques. "Superframe Duration Allocation Schemes to Improve the Throughput of Cluster-tree Networks based on IEEE 802.15.4 and ZigBee Standards.", *Sensors*, MDPI, 2016. (**UNDER CONSIDERATION**).

Erico Leão, Ricardo Moraes, Carlos Montez, Paulo Portugal, and Francisco Vasques. "CT-SIM: A Simulation Model for Wide-Scale Cluster-Tree Networks based on the IEEE 802.15.4 and ZigBee Standards.", *International Journal of Distributed Sensor Networks*, SAGE, 2016. (**UNDER CONSIDERATION**).

A.2 Published Papers

A.2.1 Book Chapters

Leão, E. M. & Vasques, F. & Portugal, P. Supporting Real-Time Communication in Large-Scale Wireless Sensor Networks. *Encyclopedia of Information Science and Technology*, Third Edition. IGI Global, pp. 2015. 7371-7380. DOI: 10.4018/978-1-4666-5888-2.ch725.

A.2.2 Publications in Conferences

Erico Leão, Francisco Vasques, Paulo Portugal, Ricardo Moraes and Carlos Montez. *An Allocation Scheme for IEEE 802.15.4-ZigBee Cluster-tree Networks*. In 42nd Annual

Conference of IEEE Industrial Electronics Society (IECON), 2016. IEEE Conference. **(ACCEPTED FOR PUBLICATION)**.

Erico Leão, Francisco Vasques, Paulo Portugal and Carlos Montez. *An Opportunistic Approach to Deal with Real-Time Mesh Communication in Wireless Sensor Networks*. In 12th IEEE International Conference on Industrial Informatics (INDIN), 2014. IEEE Conference, pp. 100-105, 27-30 July 2014. DOI: 10.1109/INDIN.2014.6945491.

References

- [1] Vehbi C Gungor and Gerhard P Hancke 0001. Industrial Wireless Sensor Networks - Challenges, Design Principles, and Technical Approaches. *IEEE Trans. Industrial Electronics*, 56(10):4258–4265, 2009. (Cited on pages 1 and 7.)
- [2] A Willig. Recent and Emerging Topics in Wireless Industrial Communications: A Selection. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, 2008. (Cited on pages 1, 2, 7 and 15.)
- [3] A Willig, K Matheus, and A Wolisz. Wireless Technology in Industrial Networks. *Proceedings of the IEEE*, 93(6):1130–1151, 2005. (Cited on page 1.)
- [4] M Mehdi Afsar and Mohammad-H Tayarani-N. Clustering in Sensor Networks. *Journal of Network and Computer Applications*, 46(C):198–226, November 2014. (Cited on pages 1, 2 and 4.)
- [5] John A Stankovic. When Sensor and Actuator Networks Cover the World. *ETRI Journal*, 30(5):627–633, October 2008. (Cited on pages 1, 3 and 50.)
- [6] Paolo Baronti, Prashant Pillai, Vince W C Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. Wireless Sensor Networks: A Survey on the State of the Art and the 802.15.4 and ZigBee standards. *Computer Communications*, 30(7):1655–1695, May 2007. (Cited on pages 2, 3, 8 and 90.)
- [7] A Ajith S Kumar, Knut Ovsthus, and Lars M Kristensen. An Industrial Perspective on Wireless Sensor Networks - A Survey of Requirements, Protocols, and Challenges. *IEEE Communications Surveys and Tutorials*, 16(3):1391–1412, 2014. (Cited on pages 2, 7 and 50.)
- [8] Muhammad Adeel Mahmood, Winston K G Seah, and Ian Welch. Reliability in Wireless Sensor Networks: A Survey and Challenges Ahead. *Computer Networks*, 79:166–187, March 2015. (Cited on pages 1, 2 and 9.)
- [9] I F Akyildiz, W Su, Y Sankarasubramaniam, and E Cayirci. Wireless Sensor Networks: a Survey. *Computer Networks*, 38(4):393–422, March 2002. (Cited on pages 2, 7, 8 and 19.)
- [10] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless Sensor Network Survey. *Computer Networks*, 52(12):2292–2330, August 2008. (Cited on pages 2, 8, 50 and 90.)
- [11] Adel Ali Ahmed and Norsheila Fisal. A Real-Time Routing Protocol with Load Distribution in Wireless Sensor Networks. *Computer Communications*, 31(14):3190–3203, September 2008. (Cited on pages 2, 17 and 20.)
- [12] Ian F Akyildiz and Ismail H Kasimoglu. Wireless Sensor and Actor Networks - Research Challenges. *Ad Hoc Networks*, 2(4):351–367, 2004. (Cited on page 3.)

- [13] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, May 2009. (Cited on page 3.)
- [14] Ameer Ahmed Abbasi and Mohamed Younis. A Survey on Clustering Algorithms for Wireless Sensor Networks. *Computer Communications*, 30(14-15):2826–2841, October 2007. (Cited on pages 3, 21 and 94.)
- [15] A A Aziz, Y A Sekercioglu, P Fitzpatrick, and M Ivanovich. A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks. *IEEE Communications Surveys and Tutorials*, 15(1):121–144, 2013. (Cited on page 3.)
- [16] Jin-Woo Kim, José Barrado, and Dong-Keun Jeon. Time and Energy Efficient Relay Transmission for Multi-Hop Wireless Sensor Networks. *Sensors*, 16(7):985, June 2016. (Cited on pages 4 and 10.)
- [17] J A Stankovic, T F Abdelzaher, Chenyang Lu, Lui Sha, and J C Hou. Real-Time Communication and Coordination in Embedded Sensor Networks. *Proceedings of the IEEE*, 91(7):1002–1022, 2003. (Cited on pages 4 and 17.)
- [18] IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs). *IEEE Computer Society*, pages 1–709, December 2015. (Cited on pages 4, 24 and 50.)
- [19] IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, 2012. (Cited on pages 4 and 15.)
- [20] IEEE Recommended Practice for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.5-2009*, pages 1–166, 2009. (Cited on pages 4, 15 and 17.)
- [21] ZigBee Specification. ZigBee Alliance (Document 053474r20), September 2012. (Cited on pages 4, 24, 25, 32, 50, 56, 90, 94 and 98.)
- [22] WirelessHART Communication Protocol Specification (IEC 62591). *HART Communication Foundation*, April 2010. (Cited on page 4.)
- [23] G Montenegro, N Kushalnagar, J Hui, and D Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *IETF, The Internet Engineering Task Force*, September 2007. (Cited on page 4.)
- [24] Ian F Akyildiz, Xudong Wang, and Weilin Wang. Wireless Mesh Networks: a Survey. *Computer Networks*, 47(4):445–487, March 2005. (Cited on page 5.)
- [25] Panneer Muthukumaran, Rodolfo de Paz Alberola, Rostislav Spinar, and Dirk Pesch. MeshMAC: Enabling Mesh Networking over IEEE 802.15.4 through Distributed Beacon Scheduling. *Ad Hoc Networks*, 28(Chapter 38):561–575, 2009. (Cited on pages 5, 50 and 90.)

- [26] Anis Koubaa, Andre Cunha, Mário Alves, and Eduardo Tovar. TDBS: A Time Division Beacon Scheduling Mechanism for ZigBee Cluster-Tree Wireless Sensor Networks. *Real-Time Systems*, 40(3):321–354, December 2008. (Cited on pages 5, 9, 32, 34, 50, 51, 90, 91, 94, 96 and 98.)
- [27] Meng-Shiuan Pan and Yu-Chee Tseng. Quick Convergecast in ZigBee Beacon-enabled Tree-based Wireless Sensor Networks. *Computer Communications*, 31(5):999–1011, 2008. (Cited on pages 5, 50 and 90.)
- [28] Mario Di Francesco, Giuseppe Anastasi, Marco Conti, Sajal K Das, and Vincenzo Neri. Reliability and Energy-Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach. *IEEE Journal on Selected Areas in Communications*, 29(8):1508–1524, 2011. (Cited on pages 5, 10, 50, 55, 79 and 90.)
- [29] Hai Lin, Lusheng Wang, and Ruoshan Kong. Energy Efficient Clustering Protocol for Large-Scale Sensor Networks. *IEEE Sensors Journal*, 15(12):7150–7160, 2015. (Cited on pages 5 and 9.)
- [30] Changle Li, Hanxiao Zhang, Binbin Hao, and Jiandong Li. A Survey on Routing Protocols for Large-Scale Wireless Sensor Networks. *Sensors*, 11(12):3498–3526, December 2011. (Cited on pages 5, 10, 16, 20, 21, 50 and 94.)
- [31] Seong-Eun Yoo, Poh Kit Chong, Daeyoung Kim, Yoonmee Doh, Minh-Long Pham, Eunjang Choi, and Jaedoo Huh. Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4. *IEEE Transactions on Industrial Electronics*, 57(11):3868–3876, 2010. (Cited on page 7.)
- [32] Habib M Ammari, Nicholas Gomes, Matthew Jacques, Bruce R Maxim, and David H Yoon. A Survey of Sensor Network Applications and Architectural Components. *Ad Hoc & Sensor Wireless Networks*, 25(1-2):1–44, 2015. (Cited on pages 7 and 8.)
- [33] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. Wireless sensor networks: a survey on recent developments and potential synergies. *Journal of Supercomputing*, 68(1):1–48, April 2014. (Cited on page 7.)
- [34] Sang Hyuk Lee, Soobin Lee, Heecheol Song, and Hwang Soo Lee. Wireless sensor network design for tactical military applications : Remote large-scale environments. In *IEEE Conference Military Communications*, pages 1–7. IEEE, 2009. (Cited on page 7.)
- [35] Xingjian Ding, Guodong Sun, Gaoxiang Yang, and Xinna Shang. Link Investigation of IEEE 802.15.4 Wireless Sensor Networks in Forests. *Sensors*, 16(7):987, June 2016. (Cited on page 7.)
- [36] Mohd Fauzi Othman and Khairunnisa Shazali. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *Procedia Engineering*, 41:1204–1210, 2012. (Cited on pages 7 and 8.)
- [37] Aqeel-ur-Rehman, Abu Zafar Abbasi, Noman Islam, and Zubair Ahmed Shaikh. A Review of Wireless Sensors and Networks' Applications in Agriculture. *Computer Standards & Interfaces*, 36(2):263–270, February 2014. (Cited on page 8.)

- [38] Cheick Tidjane Kone, Abdelhakim Hafid, and Mustapha Boushaba. Performance Management of IEEE 802.15.4 Wireless Sensor Network for Precision Agriculture. *IEEE Sensors Journal*, 15(10):5734–5747, 2015.
- [39] Mare Srbinovska, Cvetan Gavrovski, Vladimir Dimcev, Aleksandra Krkoleva, and Vesna Borozan. Environmental Parameters Monitoring in Precision Agriculture Using Wireless Sensor Networks. *Journal of Cleaner Production*, 88:297–307, February 2015. (Cited on page 8.)
- [40] Bushra Rashid and Mubashir Husain Rehmani. Applications of Wireless Sensor Networks for Urban Areas: A survey. *Journal of Network and Computer Applications*, 60:192–219, January 2016. (Cited on page 8.)
- [41] Kapileswar Nellore and Gerhard P Hancke. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. *Sensors*, 16(2), February 2016. (Cited on page 8.)
- [42] Petr Jurčík, Anis Koubaa, Ricardo Severino, Mário Alves, and Eduardo Tovar. Dimensioning and Worst-Case Analysis of Cluster-Tree Sensor Networks. *ACM Transactions on Sensor Networks*, 7(2), August 2010. (Cited on pages 9, 10, 54 and 91.)
- [43] WoongChul Choi and SeokMin Lee. A Novel GTS Mechanism for Reliable Multihop Transmission in the IEEE 802.15.4 Network. *International Journal of Distributed Sensor Networks*, 2012, 2012. (Cited on pages 9, 10, 91 and 95.)
- [44] Francesca Cuomo, Emanuele Cipollone, and Anna Abbagnale. Performance Analysis of IEEE 802.15.4 Wireless Sensor Networks: An Insight into the Topology Formation Process. *Computer Networks*, 53(18):3057–3075, 2009. (Cited on pages 9 and 51.)
- [45] Meng-Shiuan Pan, Chia-Hung Tsai, and Yu-Chee Tseng. The Orphan Problem in ZigBee Wireless Networks. *IEEE Transactions on Mobile Computing*, 8(11):1573–1584, November 2009. (Cited on page 9.)
- [46] Hyung-Sin Kim, Jae-Seok Bang, and Yong-Hwan Lee. Distributed Network Configuration in Large-scale Low Power Wireless Networks. *Computer Networks*, 70: 288–301, September 2014. (Cited on pages 9, 95 and 100.)
- [47] Francesca Cuomo, Anna Abbagnale, and Emanuele Cipollone. Cross-layer Network Formation for Energy-efficient IEEE 802.15.4/ZigBee Wireless Sensor Networks. *Ad Hoc Networks*, 11(2):672–686, March 2013. (Cited on page 9.)
- [48] H M N Dilum Bandara, Anura P Jayasumana, and Tissa H Illangasekare. A Top-Down Clustering and Cluster-Tree-Based Routing Scheme for Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 7:17, 2011. (Cited on pages 9, 51 and 94.)
- [49] E Toscano and L Lo Bello. Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, 8(2): 337–350, 2012. (Cited on pages 9, 91, 94 and 96.)

- [50] Chi-Ming Wong, Chih-Fong Chang, and Bih-Hwang Lee. A Simple Time Shift Scheme for Beacon Broadcasting Based on Cluster-Tree IEEE 802.15.4 Low-Rate WPANs. *Wireless Personal Communications*, 72(4):2837–2848, October 2013. (Cited on page 9.)
- [51] A Ahmad and Z Hanzalek. ZigBee Cluster Tree Formation for Time-bounded Data Flows in one Collision Domain. In *IEEE World Conference on Factory Communication Systems*, pages 1–4. IEEE, 2015. (Cited on pages 9 and 33.)
- [52] Z Hanzalek and P Jurcík. Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee. *IEEE Transactions on Industrial Informatics*, 6(3):438–450, 2010. (Cited on pages 9, 17, 21, 51, 55, 91, 94 and 96.)
- [53] Ricardo Severino, Nuno Pereira, and Eduardo Tovar. Dynamic Cluster Scheduling for Cluster-tree WSNs. *SpringerPlus*, 3(1):493, August 2014. (Cited on pages 55, 94, 97 and 116.)
- [54] Lun-Wu Yeh and Meng-Shiuan Pan. Beacon Scheduling for Broadcast and Convergecast in ZigBee Wireless Sensor Networks. *Computer Communications*, 38: 1–12, February 2014. (Cited on pages 9, 55, 91, 96 and 97.)
- [55] Feng Xia, Jie Li, Ruonan Hao, Xiangjie Kong, and Ruixia Gao. Service Differentiated and Adaptive CSMA/CA over IEEE 802.15.4 for Cyber-Physical Systems. *Scientific World Journal*, 2013:1–12, 2013. (Cited on pages 10, 11 and 51.)
- [56] Bih-Hwang Lee, Eppy Yundra, Huai-Kuei Wu, and M Udin Harun Al Rasyid. Analysis of Superframe Duration Adjustment Scheme for IEEE 802.15.4 Networks. *EURASIP Journal on Wireless Communications and Networking*, 2015(1):1–17, April 2015. (Cited on pages 10, 11, 51, 53, 55 and 91.)
- [57] Hadi Rasouli, Yousef S Kavian, and Habib F Rashvand. ADCA: Adaptive Duty Cycle Algorithm for Energy Efficient IEEE 802.15.4 Beacon-Enabled Wireless Sensor Networks. *IEEE Sensors Journal*, 14(11):3893–3902, 2014. (Cited on pages 10 and 55.)
- [58] J Misić. On Slave-Slave Bridging with Non-Acknowledged GTS Access in 802.15.4 Beacon Enabled Networks. In *21st International Conference on Advanced Information Networking and Applications Workshops*, pages 641–646. IEEE, 2007. (Cited on pages 10 and 95.)
- [59] P Jurcík, R Severino, A Koubaa, M Alves, and E Tovar. Real-Time Communications Over Cluster-Tree Sensor Networks with Mobile Sink Behaviour. In *14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 401–412. IEEE, 2008. (Cited on page 10.)
- [60] Xuxun Liu. A Survey on Clustering Routing Protocols in Wireless Sensor Networks. *Sensors*, 12(12):11113–11153, December 2012. (Cited on pages 10, 16, 20, 21 and 94.)

- [61] Asis Kumar Tripathy and Suchismita Chinara. Comparison of Residual Energy-Based Clustering Algorithms for Wireless Sensor Network. *ISRN Sensor Networks*, 2012(4): 1–10, 2012. (Cited on pages 10 and 94.)
- [62] Nessrine Chakchouk. A Survey on Opportunistic Routing in Wireless Communication Networks. *IEEE Communications Surveys and Tutorials*, 17(4):2214–2241, 2015. (Cited on page 10.)
- [63] Marjan Radi, Behnam Dezfouli, Kamalrulnizam Abu Bakar, and Malrey Lee. Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. *Sensors*, 12(12):650–685, December 2012. (Cited on pages 10, 16 and 20.)
- [64] Ali Al-Humairi and Alexander Probst. Back-Off Time Calculation Algorithms in WSN. *Informatics*, 3(2):1–24, June 2016. (Cited on pages 11 and 51.)
- [65] Wan Du, David Navarro, and Fabien Mieleville. Performance evaluation of IEEE 802.15.4 sensor networks in industrial applications. *International Journal of Communication Systems*, 28(10):1657–1674, 2015. (Cited on pages 11, 51 and 53.)
- [66] Erico Leão, Francisco Vasques, and Paulo Portugal. Supporting Real-Time Communication in Large-Scale Wireless Sensor Networks. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 7371–7380. IGI Global. (Cited on page 12.)
- [67] Feng Chen, Reinhard German, and Falko Dressler. Towards IEEE 802.15.4e: A study of performance aspects. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 68–73. IEEE, 2010. (Cited on pages 15 and 17.)
- [68] IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, 2011. (Cited on pages 15, 16, 56, 90, 94 and 98.)
- [69] David Rodenas-Herraiz, Antonio-Javier Garcia-Sanchez, Felipe Garcia-Sanchez, and Joan Garcia-Haro. Current Trends in Wireless Mesh Sensor Networks: A Review of Competing Approaches. *Sensors*, 13(5):5958–5995, May 2013. (Cited on pages 16 and 17.)
- [70] Kemal Akkaya and Mohamed Younis. A Survey on Routing Protocols for Wireless Sensor Networks. *Ad Hoc Networks*, 3(3):325–349, May 2005. (Cited on pages 16 and 20.)
- [71] A Koubaa, A Cunha, and M Alves. A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks. In *19th Euromicro Conference on Real-Time Systems*, pages 125–135, 2007. (Cited on pages 17 and 21.)
- [72] Z T Ayoub, S Ouni, and F Kamoun. Global versus Local Re-Association Approach to Extend the Lifetime of Large-Scale IEEE 802.15.4/ZigBee Wireless Sensor Networks. In *5th International Conference on New Technologies, Mobility and Security*, pages 1–5, 2012.

- [73] Feng Chen, T Talanis, R German, and F Dressler. Real-time Enabled IEEE 802.15.4 Sensor Networks in Industrial Automation. In *IEEE International Symposium on Industrial Embedded Systems*, pages 136–139. IEEE, 2009. (Cited on pages 17, 18 and 19.)
- [74] Wun-Cheol Jeong and Junhee Lee. Performance Evaluation of IEEE 802.15.4e DSME MAC Protocol for Wireless Sensor Networks. In *First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things*, pages 7–12. IEEE, 2012. (Cited on page 17.)
- [75] David Stanislawski, Xavier Vilajosana, Qin Wang, Thomas Watteyne, and Kris Pister. Adaptive Synchronization in IEEE802.15.4e Networks. *IEEE Transactions on Industrial Informatics*, (99):1–1, March 2013. (Cited on page 17.)
- [76] M J Lee, Rui Zhang, Jianliang Zheng, Gahng-Seop Ahn, Chunhui Zhu, Tae Park, Sung Rae Cho, Chang Sub Shin, and Jun Sun Ryu. IEEE 802.15.5 WPAN Mesh Standard-Low Rate Part: Meshing the Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*, 28(7):973–983, 2010. (Cited on page 17.)
- [77] Kideok Cho, Munyoung Lee, Dong-il Ko, Taekyoung Kwon, and Yanghee Choi. An Efficient Multicast Routing in IEEE 802.15.5 Networks. In *International Conference on Information Networking*, pages 1–3, 2009. (Cited on page 17.)
- [78] Z Teng and K I Kim. A Survey on Real-time MAC Protocols in Wireless Sensor Networks. *Communications and Network*, 2:104–112, May 2010. (Cited on pages 17 and 18.)
- [79] M I Channa and I Memon. Real Time Traffic Support in Wireless Sensor Networks. In *Conference on Engineering Sciences and Technology*, pages 1–6, 2005. (Cited on page 18.)
- [80] Feng Xia, Ruonan Hao, Yang Cao, and Lei Xue. A Survey of Adaptive and Real-Time Protocols Based on IEEE 802.15.4. *International Journal of Distributed Sensor Networks*, 2011(2):1–11, 2011. (Cited on page 19.)
- [81] Mohammadreza Eslaminejad and Shukor Abd Razak. Fundamental Lifetime Mechanisms in Routing Protocols for Wireless Sensor Networks: A Survey and Open Issues. *Sensors*, 12(12):13508–13544, December 2012. (Cited on page 20.)
- [82] J N Al-Karaki and A E Kamal. Routing Techniques in Wireless Sensor Networks: a Survey. *IEEE Wireless Communications*, 11(6):6–28, 2004. (Cited on page 20.)
- [83] Mohammad Hossein Anisi, Adbul Hanan Abdullah, Shukor Abd Razak, and Asri Ngadi. Overview of Data Routing Approaches for Wireless Sensor Networks. *Sensors*, 12(12): 3964–3996, December 2012. (Cited on page 20.)
- [84] W R Heinzelman, A Chandrakasan, and H Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *International Symposium on Pervasive Systems, Algorithms and Networks*, page 10. IEEE Comput. Soc, 2000. (Cited on pages 20 and 94.)

- [85] O Boyinbode, Hanh Le, A Mbogho, M Takizawa, and R Poliah. A Survey on Clustering Algorithms for Wireless Sensor Networks. In *13th International Conference on Network-Based Information Systems*, pages 358–364, 2010. (Cited on page 21.)
- [86] John A Stankovic, Anthony D Wood, and Tian He. Realistic Applications for Wireless Sensor Networks. *Theoretical Aspects of Distributed Computing in Sensor Networks*, (Chapter 25):835–863, 2011. (Cited on pages 23 and 50.)
- [87] Miodrag Živković, Boško Nikolić, Jelica Protić, and Ranko Popović. A Survey and Classification of Wireless Sensor Networks Simulators Based on the Domain of Use. *Adhoc & Sensor Wireless Networks*, 20(3/4):1–43, July 2014. (Cited on pages 24, 26, 28, 73 and 110.)
- [88] Muhammad Imran, Abas Md Said, and Halabi Hasbullah. A Survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks. In *International Symposium on Information Technology*, pages 897–902. IEEE, 2010. (Cited on pages 24 and 28.)
- [89] A K Dwivedi. An Exploratory Study of Experimental Tools for Wireless Sensor Networks. *Wireless Sensor Network*, 03(07):215–240, 2011. (Cited on pages 24, 26 and 28.)
- [90] Dimosthenis Pediaditakis, Yuri Tselishchev, and Athanassios Boulis. Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator. In *3rd International ICST Conference on Simulation Tools and Techniques*, page 53. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), March 2010. (Cited on page 24.)
- [91] M Korkalainen, M Sallinen, N Karkkainen, and P Tukeva. Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications. In *Fifth International Conference on Networking and Services*, pages 102–106, 2009. (Cited on pages 26 and 28.)
- [92] Laura Marie Feeney. Towards Trustworthy Simulation of Wireless MAC/PHY Layers: a Comparison Framework. In *15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, page 295, New York, New York, USA, October 2012. ACM Request Permissions. (Cited on page 26.)
- [93] M Z Khan, B Askwith, F Bouhafs, and M Asim. Limitations of Simulation Tools for Large-Scale Wireless Sensor Networks. In *IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pages 820–825, 2011. (Cited on pages 26, 27 and 28.)
- [94] Ivan Stojmenovic. Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions. *IEEE Communications Magazine*, 46(12): 102–107, December 2008. (Cited on page 27.)
- [95] Philip Levis, Nelson Lee, Matt Welsh, and David E Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *1st International Conference on Embedded Networked Sensor Systems*, pages 126–137, New York, New York, USA, 2003. ACM Press. (Cited on page 28.)

- [96] Javier Hurtado-López, Eduardo Casilari, and A Ariza-Quintana. Enabling IEEE 802.15.4 Cluster-Tree Topologies in OMNeT++. In *2nd International Conference on Simulation Tools and Techniques*. ICST, March 2009. (Cited on page 29.)
- [97] F Chen and F Dressler. A Simulation Model of IEEE 802.15. 4 in OMNeT+. Technical report, 2007. (Cited on page 29.)
- [98] Erico Leão, F Vasques, P Portugal, and C Montez. An Opportunistic Approach to Deal with Real-Time Mesh Communication in Wireless Sensor Networks. In *12th IEEE International Conference on Industrial Informatics*, pages 100–105. IEEE, 2014. (Cited on pages 30 and 36.)
- [99] Mitchel S Felske, Carlos Montez, Alex S R Pinto, Francisco Vasques, and Paulo Portugal. GLHOVE: A Framework for Uniform Coverage Monitoring using Cluster-Tree Wireless Sensor Networks. In *IEEE 18th Conference on Emerging Technologies & Factory Automation*, pages 1–8. IEEE, 2013. (Cited on pages 30, 53, 61, 66, 95, 100, 116 and 124.)
- [100] Erico Leão, Francisco Vasques, Paulo Portugal, Ricardo Moraes, and Carlos Montez. An Allocation Scheme for IEEE 802.15.4-ZigBee Cluster-tree Networks . In *42nd Annual Conference of IEEE Industrial Electronics Society*, pages 1–6, 2016. (Cited on pages 34, 36, 41 and 51.)
- [101] F Vasques and G Juanole. Pre-run-time schedulability analysis in Fieldbus networks. In *IECON'94 - 20th Annual Conference of IEEE Industrial Electronics*, pages 1200–1204. IEEE, 1994. (Cited on page 51.)
- [102] Nicholas Malcolm and Wei Zhao. The Timed-Token Protocol for Real-Time Communications. *IEEE Computer*, 27(1):35–41, 1994. (Cited on page 51.)
- [103] R Lange, F Vasques, P Portugal, and R S de Oliveira. Guaranteeing real-time message deadlines in the FlexRay static segment using a on-line scheduling approach. In *9th IEEE International Workshop on Factory Communication Systems*, pages 301–310. IEEE, 2012. (Cited on page 51.)
- [104] Rodrigo Lange, Francisco Vasques, Rômulo Silva de Oliveira, and Paulo Portugal. A scheme for slot allocation of the FlexRay Static Segment based on response time analysis. *Computer Communications*, 63:65–76, 2015. (Cited on pages 51 and 64.)
- [105] Xianghui Cao, Jiming Chen, Yu Cheng, Xuemin Sherman Shen, and Youxian Sun. An Analytical MAC Model for IEEE 802.15.4 Enabled Wireless Networks With Periodic Traffic. *IEEE Transactions on Wireless Communications*, 14(10):5261–5273, October 2015. (Cited on page 53.)
- [106] Mouhcine Guennoun, Mounib Khanafer, and Hussein T Mouftah. Modeling of variable Clear Channel Assessment MAC protocol for Wireless Sensor Networks. *Computer Communications*, 59(C):67–83, March 2015. (Cited on page 54.)
- [107] Joaquin Recas, Nadia Khaled, Alberto A Del Barrio, and Roman Hermida. Generic Markov model of the contention access period of IEEE 802.15.4 MAC layer. *Digital Signal Processing*, 33:191–205, October 2014. (Cited on page 54.)

- [108] Pangun Park, Carlo Fischione, and Karl Henrik Johansson. Modeling and Stability Analysis of Hybrid Multiple Access in the IEEE 802.15.4 Protocol. *ACM Transactions on Sensor Networks*, 9(2):–55, March 2013. (Cited on pages 53 and 54.)
- [109] Marco Martalò, Stefano Busanelli, and Gianluigi Ferrari. Markov Chain-based performance analysis of multihop IEEE 802.15.4 wireless networks. *Performance Evaluation*, 66(12):722–741, December 2009. (Cited on page 54.)
- [110] Mikko Kohvakka, Mauri Kuorilehto, Marko Hännikäinen, and Timo D Hämäläinen. Performance analysis of IEEE 802.15.4 and ZigBee for large-scale wireless sensor network applications. In *3rd ACM International Workshop on Performance Evaluation of Wireless Ad hoc, Sensor and Ubiquitous Networks*. ACM Request Permissions, 2006. (Cited on pages 54, 62 and 63.)
- [111] Feng Chen, Nan Wang, Reinhard German, and Falko Dressler. Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications. *Wireless Communications and Mobile Computing*, 10(5):609–621, 2009. (Cited on pages 55 and 79.)
- [112] E Casilari, J Hurtado-dueñas, and J M Cano-garcía. A study of policies for beacon scheduling in 802.15. 4 cluster-tree networks. In *9th International Conference on Applied Computer Science*, pages 124–129, 2009. (Cited on pages 55 and 56.)
- [113] P Suriyachai, U Roedig, and A Scott. A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks. *IEEE Communications Surveys and Tutorials*, 14(2):240–264, 2012. (Cited on page 60.)
- [114] C L Liu and James W Layland. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of the ACM*, 20(1):46–61, January 1973. (Cited on page 61.)
- [115] Gopal Agrawal, Biao Chen, Wei Zhao, and Sadegh Davari. Guaranteeing Synchronous Message Deadlines with the Timed Token Medium Access Control Protocol. *IEEE Transactions on Computers*, 43(3):327–339, 1994. (Cited on page 64.)
- [116] N Audsley, A Burns, M Richardson, K Tindell, and A J Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5):284–292, 1993. (Cited on page 64.)
- [117] Yuriy Tselishchev, Athanassios Boulis, and Lavy Libman. Experiences and Lessons from Implementing a Wireless Sensor Network MAC Protocol in the Castalia Simulator. *IEEE Wireless Communication and Networking Conference*, pages 1–6, 2010. (Cited on pages 73 and 110.)
- [118] S J Shobana and B Paramasivan. CRAP: Cluster based Congestion Control with Rate Adjustment based on Priority in Wireless Sensor Networks. *International Journal of Multimedia and Ubiquitous Engineering*, 10:421–436, 2015. (Cited on page 91.)
- [119] Dilip Kumar, Trilok C Aseri, and R B Patel. EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks. *Computer Communications*, 32(4): 662–667, March 2009. (Cited on page 94.)

- [120] Wenjuan Liu, Dongmei Zhao, and Gang Zhu. End-to-end delay and packet drop rate performance for a wireless sensor network with a cluster-tree topology. *Wireless Communications and Mobile Computing*, 14(7):729–744, 2014. (Cited on page 94.)
- [121] Junghee Han, Suhan Choi, and Taejoon Park. Maximizing lifetime of cluster-tree ZigBee networks under end-to-end deadline constraints. *IEEE Communications Letters*, 14(3): 214–216, 2010. (Cited on page 94.)
- [122] Vinay Kumar, Sanjeev Jain, and Sudarshan Tiwari. Energy Efficient Clustering Algorithms in Wireless Sensor Networks: A Survey. *International Journal of Computer Science Issues*, 8(5):259–268, September 2011. (Cited on page 94.)
- [123] Soroush Naeimi, Hamidreza Ghafghazi, Chee-Onn Chow, and Hiroshi Ishii. A Survey on the Taxonomy of Cluster-Based Routing Protocols for Homogeneous Wireless Sensor Networks. *Sensors*, 12(6):7350–7409, May 2012. (Cited on page 94.)
- [124] Chuan Zhu, Shuai Wu, Guangjie Han, Lei Shu, and Hongyi Wu. A Tree-Cluster-Based Data-Gathering Algorithm for Industrial WSNs With a Mobile Sink. *IEEE Access*, 3: 381–396, 2015. (Cited on page 95.)
- [125] A Khatiri, G Mirjalily, and A Khademzadeh. Energy-Efficient Shortcut Tree Routing in ZigBee Networks. In *Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 117–122. IEEE, 2012. (Cited on page 95.)
- [126] Yu-Kai Huang, Ai-Chun Pang, Pi-Cheng Hsiu, Weihua Zhuang, and Pangfeng Liu. Distributed Throughput Optimization for ZigBee Cluster-Tree Networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(3):513–520, March 2012. (Cited on page 95.)
- [127] N Abdeddaim, F Theoleyre, F Rousseau, and A Duda. Multi-Channel Cluster Tree for 802.15.4 Wireless Sensor Networks. In *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications*, pages 590–595, 2012. (Cited on page 96.)
- [128] E W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, December 1959. (Cited on page 102.)
- [129] Farrokh Etezadi, Keyvan Zarifi, Ali Ghayeb, and Sofiene Affes. Decentralized Relay Selection Schemes in Uniformly Distributed Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 11(3):938–951, March 2012. (Cited on page 103.)
- [130] Lingya Liu, Cunqing Hua, Cailian Chen, and Xinping Guan. Relay Selection for Three-Stage Relaying Scheme in Clustered Wireless Networks. *IEEE Transactions on Vehicular Technology*, 64(6):2398–2408, June 2015. (Cited on page 103.)
- [131] D G Costa, I Silva, L A Guedes, F Vasques, and P Portugal. Availability Issues in Wireless Visual Sensor Networks. *Sensors*, 2014. (Cited on page 114.)